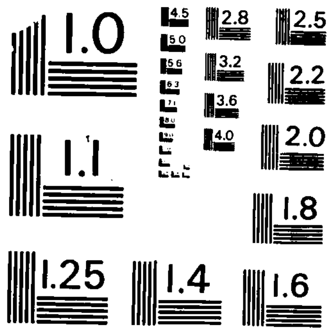
AD-A175 137 COMPUTERIZED COST OPTIMIZATION SCHEDULING(U) ARMY MILITARY PERSONNEL CENTER ALEXANDRIA VA M A RONCOLI 21 NOV 86 1/2

UNCLASSIFIED F/G 14/1 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(2)

AD-A175 137

COMPUTERIZED COST OPTIMIZATION SCHEDULING

BY

MARK A RONCOLI

A REPORT PRESENTED TO THE GRADUATE COMMITTEE
OF THE DEPARTMENT OF CIVIL ENGINEERING IN
PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF ENGINEERING

UNIVERSITY OF FLORIDA

Fall 1986

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (*and Subtitle*)

COMPUTERIZED COST OPTIMIZATION SCHEDULING

5. TYPE OF REPORT & PERIOD COVERED

21 NOV 86

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

MARK A RONCOLI

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS

STUDENT, HQDA, MILPERCEN (DAPC-OPA-E), 200 STOVALL STREET, ALEXANDRIA, VIRGINIA, 22332

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

HQDA, MILPERCEN DAPC-OPA-E, 200 STOVALL STREET, ALEXANDRIA, VIRGINIA, 22332

12. REPORT DATE

21 NOV 86

13. NUMBER OF PAGES

90

14. MONITORING AGENCY NAME & ADDRESS (*If different from Controlling Office*)

15. SECURITY CLASS. (*of this report*)

N/A

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (*of this Report*)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)

18. SUPPLEMENTARY NOTES

DOCUMENT IS A MASTER'S REPORT

DOCUMENT [illegible] CONTAINS A LISTING OF COMPUTER TEXT FOR COST OPTIMIZATION SCHEDULING AS WRITTEN BY THE AUTHOR.

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)

COST OPTIMIZATION SCHEDULING; COMPUTERIZED APPROACH; PROJECT [illegible]; CPM

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

THE SCHEDULING OF PROJECTS IS A VITAL FUNCTION FOR THOSE IN THE CONSTRUCTION INDUSTRY. ANY SCHEDULING STRATEGY THAT PROVIDES A MEANS FOR REDUCING PROJECT COST [illegible] OF [illegible] TO A CONTRACTOR, PERFORMING [illegible] [illegible] [illegible] (AND PROVIDES CONTRACTORS) WITH

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

THIS VALUABLE MANAGEMENT INFORMATION. UTILIZING THE ABILITIES OF THE COMPUTER ENHANCES THE INFORMATION PROCESS. THE PURPOSE OF THIS MASTER'S REPORT IS TO PRESENT THE CONCEPTS CONCERNING COST OPTIMIZATION SCHEDULING AND A COMPUTERIZED PROGRAM THAT WAS DEVELOPED.

Computerized Cost Optimization Scheduling

CPT Mark A Roncult
HQDA, MILPERCEN (DAPC-OPA-E)
200 Stovall Street

21 November 1986

Approved for public release; distribution unlimited.

[illegible] report submitted to the University of Florida, [illegible] in partial fulfillment of the requirements for the [illegible] of Engineering.

COMPUTERIZED COST OPTIMIZATION SCHEDULING

BY

MARK A RONCOLI

A REPORT PRESENTED TO THE GRADUATE COMMITTEE
OF THE DEPARTMENT OF CIVIL ENGINEERING IN
PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF ENGINEERING

UNIVERSITY OF FLORIDA

Fall 1986

ACKNOWLEDGEMENTS

I would like to express my appreciation to a number of individuals without whose support this report would not have been possible. First, I thank the Lord Jesus Christ for His presence in my life and the insight He provided me when I was at a loss at making the compression phase of my program work. Second, I thank Joanne and the children for their wonderful support and love for me. Third, I am thankful to the US Army for providing me this opportunity to attend the University of Florida. Fourth, I thank the professors who taught the construction engineering courses for their concern and instruction. And finally, I thank Northwest Baptist Church, the Keysers and the other families in Gainesville whose support made living here so much more full.

ABSTRACT

The scheduling of projects is a vital function for those in the construction industry. Any scheduling strategy that provides a means for reducing project costs is of interest to a contractor. Performing cost optimization scheduling can provide contractors with this valuable management information. Utilizing the abilities of the computer enhances the information process. The purpose of this master's report is to present the concepts concerning cost optimization scheduling and a computerized optimization program that was developed.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

CHAPTER I
INTRODUCTION

Background

The construction industry experiences an extremely high mortality rate. Every year more than 1000 firms fail to stay in business. The industry, which accounts for nine percent of the total number of business firms in the United States, accounts for more than 17 percent of all the business failures. (10, p.39) There are a number of factors which contribute to this alarming figure, but certainly, "Almost all these causes of contractor failure can be attributed to various shortcomings on the part of management." (10, p.41) One management aspect that the contractor has to be attentive to is the scheduling of projects. This vital management function has caused the development of many various scheduling techniques. One of the most useful scheduling techniques for the past 30 years has been the Critical Path Method (CPM). In the early 1960's, this technique was utilized for the development of a process by which contractors could relate costs and minimize them. Known by a variety of names, it will be expressed in this presentation by the term cost optimization scheduling.

Current Situation

Why should a contractor, whose very existence may depend upon how well he manages his project schedules, disregard the process of cost optimization scheduling? One of the main reasons is that those in the industry have had to perform the process manually. As one author put it, "This is a calculation not suitable for the manual mode." (9, p.221) In 1984, this same author writes that the most significant barrier to usage of this approach is the lack of a computer program for a currently viable computer. He continues by stating that, "only one computer program has been available for the calculation. That one, by James E. Kelly, Jr., for the GE 225 computer, is now obsolete." (9, p.220)

Purpose

The purpose of this Master's Report is to present the concepts concerning cost optimization scheduling and to provide the construction industry with a solution to the software dilemma. Given the previous information, it is clear that there exists the need for the development of computer software with regards to the cost optimization process. While new programs may have come into existence since 1984, they are by no means common to the industry. This report attempts to fill this need with the development of a computerized cost optimization scheduling program called Crash. This software program has already been incorporated for use into one course of instruction at the

University of Florida. It is being utilized in the Civil Engineering course, ECI 5147 Construction Planning and Scheduling. It was written in the Basic computer language and then compiled to enhance its operating speed.

CHAPTER II
PROJECT EXPEDITING

Contractor Concerns

"In construction planning, the objective function is minimum cost for a specified project time." (2, p.85) Inherent in that statement is the reality that most construction contractors face. In bidding for a job, a construction contractor is presented with a required project completion time frame and must attempt to plan his work minimizing the cost of doing so. If he is successful in getting the job, then he has the opportunity to put his plan into affect. If he keeps his costs to the absolute minimums and completes the project on time or sooner, then he is sure to receive the greatest profits possible from the job. However, since in his plan the construction contractor has made assumptions about many factors including the weather, labor availability, maintenance of equipment, etc., his plan must be subject to possible alterations. Quite often the contractor will be faced with being behind in his schedule. Therefore, he faces the possibilities of damages for not meeting the specified completion date. Whether the contractor experiences the situation of the plan going smoothly or not, his ability to

utilize the concept of expediting a project is important to his profitability. In the next two paragraphs, project expediting as a function of the contractor's concern for profitability will be addressed.

Greatest Profits

"In contracting the area where profits must be made is the area between the estimated direct cost of doing a job and the amount bid for the job." (10, p.11) Given that any contractor wishes to maximize his profits, he simply needs to somehow enlarge the area between the direct costs incurred from the job and the amount he bid for the job. The problem the contractor confronts in attempting to do so is the multitude of factors which affect this area. These include his competition, amount of markup, and direct cost estimate. It is clear that if a particular contractor's estimate of direct costs are significantly lower than his competitors, then he not only stands an excellent chance of getting the job, but also should receive a significant profit. However, the previous quote was not entirely correct. There is a circumstance for increasing direct costs in an effort to maximize profits. Given that many contracts include bonuses for early completion, it is possible that a greater profit can be achieved by increasing direct costs in order to reduce project time. It may be that even the requirement of bonuses need not be present for an increase in direct costs to lead to greater profits. How can that be?

> Increased direct costs due to crash action may result in saving time, thereby increasing capital turnover, and possibly decreasing total costs as well, if the cost of the crash action is less than the incidence of indirect costs for the time saved. (4, p.145)

Therefore, the quote stating where profits come from should have described it as the difference between the amount bid and total costs. If total costs are reduced to a minimum, the contractor should realize his greatest profits. Project expediting is one technique that the contractor should have in his repetoire. It may prove profitable.

Time Reduction Need

Beyond the concept of greatest profit, there is also the need for project expediting due to a number of various contractor concerns. The need to reduce project time could be noted from the outset of construction or later on if the project is already under way. The contractor may desire to achieve completion by a certain date to avoid undesirable weather, spring run-off, or to free men and equipment for other work. Work may have to be completed in a prescribed fiscal period. The job may be under way and activities may have experienced delays, resulting in lost time that has to be recovered. For these and other reasons, the contractor occasionally finds himself faced with the situation of needing to reduce the remaining time to complete the project. He must expedite the project in order to meet his own imposed time limitation. (5, p.124)

Approaches to Earlier Project Completion

A contractor has two basic approaches to expediting a job, the first of which is to reanalyze the project schedule's network to see if there is a more efficient way of organizing the activities. Given that the estimator's calculations were accurate and the project schedule's network of activities cannot be improved upon, the only other approach for the contractor to utilize is to decrease the duration of the project's critical activities. Both of these approaches will be discussed in the following paragraphs.

Revamp the Project Network

When a reduction in project time is thought necessary, and before incurring the extra direct costs of expediting, the contractor should reexamine the critical path and the activities that make it up. The intent of this is to take advantage of any factors previously overlooked that will result in greater time efficiency without increasing the direct costs. While this operation may prove to be futile, it is definitely worth investigating. The following paragraphs will highlight four steps which may be used to revamp the project network so as to shorten the job without increasing direct costs.

Review Estimates

In reexamining the project network's critical activities, the very first step that should be taken is to review all the activity duration estimates.

Errors could have been made. For that reason, it is worthwhile to verify the original estimates. Another aspect to consider is the disparity between activity durations estimated by different people. Depending on the behavioral characteristics of the estimator, the activity durations may be generally optimistic or pessimistic. Once the pattern of bias has been established, allowances to compensate for it can be applied. Reviewing the estimates for activity durations is an obvious approach to searching for a shorter project duration. However, it should be noted that any changes made in the original estimates must be for bona fide reasons. (5, p.129)

Put Critical Activities in Parallel

Following the check for accuracy of activity durations, an examination for critical activities that could be done in parallel rather than in series should be performed. This next step explores the possibility that two or more activities could be accomplished in a simultaneous manner. It may be that the reason against such a possibility is due to a labor or equipment constraint. If a contractor could subcontract one of these activities at no increase in direct costs, then it would be quite possible to perform these critical activities in parallel. (5, p.129-130)

Subdivide Critical Activities

The third check that should be made of each critical activity is to determine if that activity must actually be completely finished prior to the next critical activity commencing. It is conceivable that a critical activity could be subdivided and a portion of it be done in parallel with another. (5, p.130) This one fact points out the major advantage of the Advanced Precedence Diagram method of CPM that will be discussed later.

Change Logic

The most drastic step to reduce project duration without increasing direct costs is the reworking of the logic for a portion of the project. This is the fresh idea approach of innovative thinking for the purpose of saving time. Established procedures are too often accepted as the only way to get a job done. Inquiring minds who believe that there may be a better way could produce excellent new approaches to the job. In effect, all sorts of tasks and constraints could be eliminated and replaced by tasks no more costly but quicker to complete. (5, p.132)

Decrease the Duration of Critical Tasks

If the reexamination of the project's network as discussed in the previous paragraphs does not reduce the critical path's duration, then the contractor has to literally "buy time". He does so by expediting critical

activities through the process of increasing the direct costs incurred to perform the tasks while decreasing their duration. There are a number of methods that a contractor can employ when faced with having to utilize this approach of project expediting. They are listed as follows:

1. Increase the number of work hours/day (overtime).
2. Utilize multiple shifts (premium night wages).
3. Bring larger and/or more equipment on the job.
4. Put more workers on the job.
5. Use more costly but quicker installed materials.

(12, p.299-300)

Each one of these methods will have the impact of increasing the direct costs of completing the project. But, as mentioned before, if the effect is a reduction in total costs, then the contractor can expect greater profits.

Strategies for Shortening Tasks

The various methods that can be employed to reduce durations of critical activities is vital for the contractor to understand. Along with the method(s) chosen is the strategic employment of the expediting procedure. The question that the contractor must answer is which one or more of the critical tasks should be identified for expediting, which is also known as compression. There are many alternative approaches that the contractor can choose from as his strategic plan for shortening critical activities. Ultimately, every one of them, as described below, will shorten the project's overall duration and, therefore, are valid as expediting strategies for a contractor's use.

Alternatives

The alternative strategies from which a contractor can choose which critical activities to identify for compression are listed and explained as follows:

1. Early activities: This gives the contractor more flexibility later on in the project.
2. Longest tasks: Generally, you can only squeeze so much out of any task. Therefore, the greatest time reduction often comes from the longest task.
3. Easiest tasks: Look for tasks that have been done before and there are fewer unknowns.
4. Those tasks for which more resources are available: It may be easier to find more plumbers than electricians.
5. Those tasks that your organization controls: Usually it is easier to control work done within your own organization.
6. Those tasks that cost the least to speed up: Generally, the most common alternative selected, because all contractors are concerned about cost.

(3, p.80-81)

Other considerations

There are yet more factors which may influence a contractor in determining which critical activities to expedite. Two such factors that will be addressed are seasonal considerations and the desire to free resources. If the project schedule calls for concrete and masonry work to be performed in a predicted time frame of freezing weather, it may prove to be economically advantageous to perform preparatory tasks on a multiple-shift basis during the preceding warmer weather so that those activities, too, can be accomplished in the warmer temperature. Depending on the local climate, seasonal considerations are of great importance to the contractor. A contractor may also wish to expedite certain activities in order to free committed

resources. By shortening the commitments on one project, the contractor can make available those resources for another job. Given these considerations and the previous alternatives from which a contractor must choose his strategy for expediting a project, it is understandable why contracting is such a tough business. There are a lot of variables involved. (12, p.298)

Least Cost Approach to Compression

As described earlier, the most often used alternative strategy for selecting tasks to be expedited is to choose those tasks which increase direct costs the least. This is often referred to as the least-cost approach to compression. It is the preferred approach largely because it does deal in terms of cost. A contractor can easily comprehend why he would want to compress the activities which would cost him the least. Procedurally, while it is manually an arduous task, he can make the calculations to identify those tasks which insure the least cost. It is this approach that will serve as the basis for all further discussion concerning project expediting in this report.

Unlimited Resource Assumption

One major assumption is made when utilizing this approach for compressing activities. It is assumed that unlimited resources are available for any requirements made by the compression of activities. In the United States, it is an assumption that is easy to digest. If the contractor needs to obtain some piece of equipment, laborers, or

materials, there is probably some price he can pay that would insure him the resource. Whether or not it would serve him advantageously would have to be seen. There are several questions the contractor would have to ask himself. Does the decrease in project time and indirect costs offset the increase in direct costs? Are there other activities that can be compressed less expensively? However, the assumption that unlimited resources are available can be made. (8, p.183)

Shortening Limitations

The least-cost approach to compression of the project cannot reduce project time down to zero. There are a number of restrictions that apply to the process of reducing the critical path. These limitations are summarized as follows:

1. Crash limit of individual activity: There is a limit to which any activity can be shortened if at all.
2. Shortening limited by float: Sometimes shortening one activity will reduce the float time of other activities causing another chain of activities to become critical and prevent further expediting.
3. Shortening limited by parallel critical paths: If one length of a parallel critical path is to be decreased, a commensurate decrease in the other length must be made. If it cannot the project duration is not reduced.
4. Shortening limited by crashed critical paths: When all of the activities of any critical path have been shortened to their full capacity, no further reduction in project time is possible. (5, p.142)

CHAPTER III
TIME-COST TRADE-OFFS

Overview of the Concept

The old phrase, "time is money", recognizes the close relationship between time and cost. A major advantage of scheduling a project utilizing CPM is the ability to find a specific relationship between time and cost for the project. It is that relationship which is the key to determining the optimum schedule, defined as the schedule or project duration that minimizes total cost. Total cost is broken down into direct and indirect costs and will be defined later in this chapter. The trade-offs are directed towards the task of determining the proper mix of the two forms of cost in order to produce the minimum sum. As was noted before, many in the construction industry assume that the best performance time for an activity is the one that minimizes direct cost. This time is usually longer than the minimum time required to carry out the activity. The strategic planners that understand this relationship and utilize the time-cost trade-offs to their advantage, while incurring greater direct costs, will minimize total costs. This chapter will focus on this concept by defining the

terms and assumptions that are involved in the process of determining cost optimization scheduling.

Various Cost Factors

Before the discussion of project time optimization can proceed further, it is necessary to understand the various factors related to the trade-offs of time and cost. The development of the entire CPM time-cost trade-off procedure is based on a number of special terms that will be defined in the next few paragraphs.

Direct Costs

The project's direct cost is the sum of the individual activities' direct costs. The direct cost of an activity is the sum of expenses of labor, equipment, materials, and subcontracts that are directly associated with the physical completion of the activity. In order to understand the time-cost relationship as it relates to direct costs, there are different terms and factors that need to be explained.

Normal Time/Cost

Each activity has a normal cost and a normal completion time. The normal cost is equal to the absolute minimum direct cost required to complete the activity. The normal time is the shortest time required to perform the activity under the minimum direct cost constraint. Performing the activity in a longer duration is considered an unnecessary "drag-out". Likewise, it makes sense that for an activity to be accomplished in less time than the minimum time

(or normal time), it would require additional costs above the normal cost.

Crash Time/Cost

The crash time is the minimum or fully expedited activity duration time that is possible. The crash cost is assumed to be the minimum direct cost required to achieve this crash duration. Any costs incurred attempting to expedite the activity further is considered to be "unnecessary crash costs". The crash cost is greater than the normal cost, while the related time is less than the normal time.

Time-Cost Slope

The differences between costs and times for the crash and normal points represents the area in which the scheduler of the project can make trade-offs. In figure 3-1, an example of an activity's time-cost trade-off curve is depicted. As shown, the actual time-cost relationship is assumed to be convex. This is due to the real world situation that to compress an activity's duration at the least cost, it takes additional dollars the more it is reduced. For example, to reduce a particular activity only one day, it may require one additional worker, but to reduce it two days may require six additional workers over the normal crew size. While there are complex compression procedures that better approximate this actual curve, they are more time consuming even with the use of the

computer and are still subject to an estimator's error. In practice, a simpler procedure of approximating this curve is taken. A linear relationship is assumed to be an adequate approximation of the actual curve. (8, p.186) Thus, the time-cost trade-off is represented by a straight line connecting the normal and crash points. The equation that represents this additional cost incurred in reducing the activity's duration one time unit below the normal time (or slope of the line) is:

$$\text{Time-Cost Slope} = \frac{(\text{Crash Cost} - \text{Normal Cost})}{(\text{Normal Time} - \text{Crash Time})}$$

(7, p.131)

Example Activity Time-Cost Trade-Off Curve

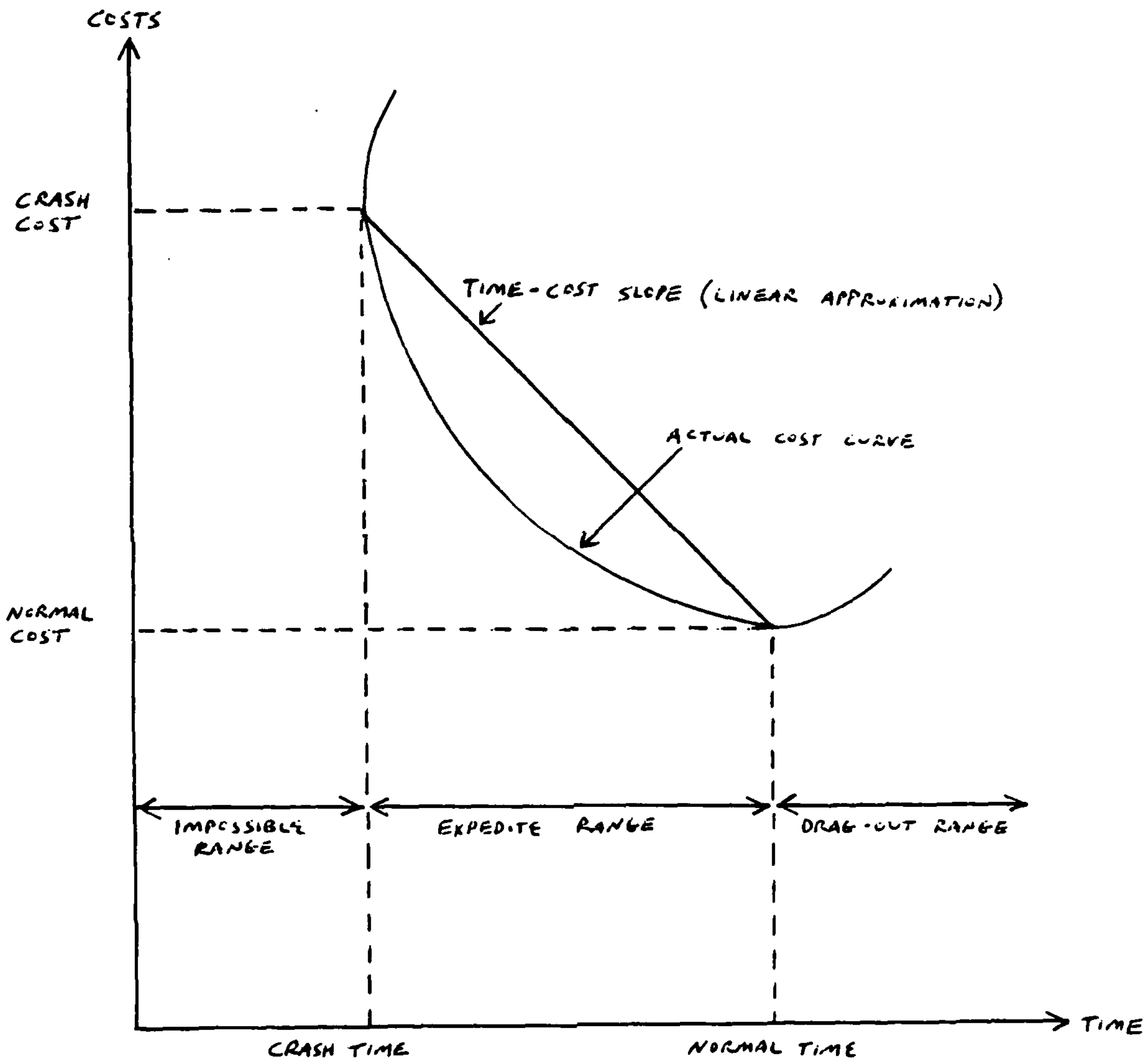


Figure 3-1

Indirect Costs

Indirect costs are those costs that cannot be associated directly with any one activity of a project. Total indirect cost can be broken down into two broad categories: general overhead and job overhead. Job overhead includes items that are affected by project duration and is the portion of indirect cost with which the planner can manage time-cost trade-offs. The next paragraphs will describe what makes up these two categories of indirect cost and how project time affects the indirect cost. (6, p.202-203)

General Overhead

General overhead is the cost of doing business. Included in this category is the salary of the company's president, the cost of the telephone, heating and lighting in the central office, the cost of supporting the accounting office, etc. It is the summation of all the costs that would be incurred even if there was no work or jobs to be done by the business. It, therefore, has no relationship with the time to do a project.

Job Overhead

This category of indirect costs account for those costs that apply to a particular project but cannot be identified with any particular activity in the job. There are a variety of types of job overheads. One type is the indirect cost incurred because of

incidentals. This includes costs that are a result of such things as building haul roads, or putting up fences, or any other activity which is used through the life of the project but is not a part of the work. These overhead costs do not vary with project duration. Another part of the job overhead category includes those project costs which are affected by the duration of the project. Included in this group are the costs of operation of a site office, supervision of the job, finance charges, damage charges for late completion, bonuses for early completion, etc.

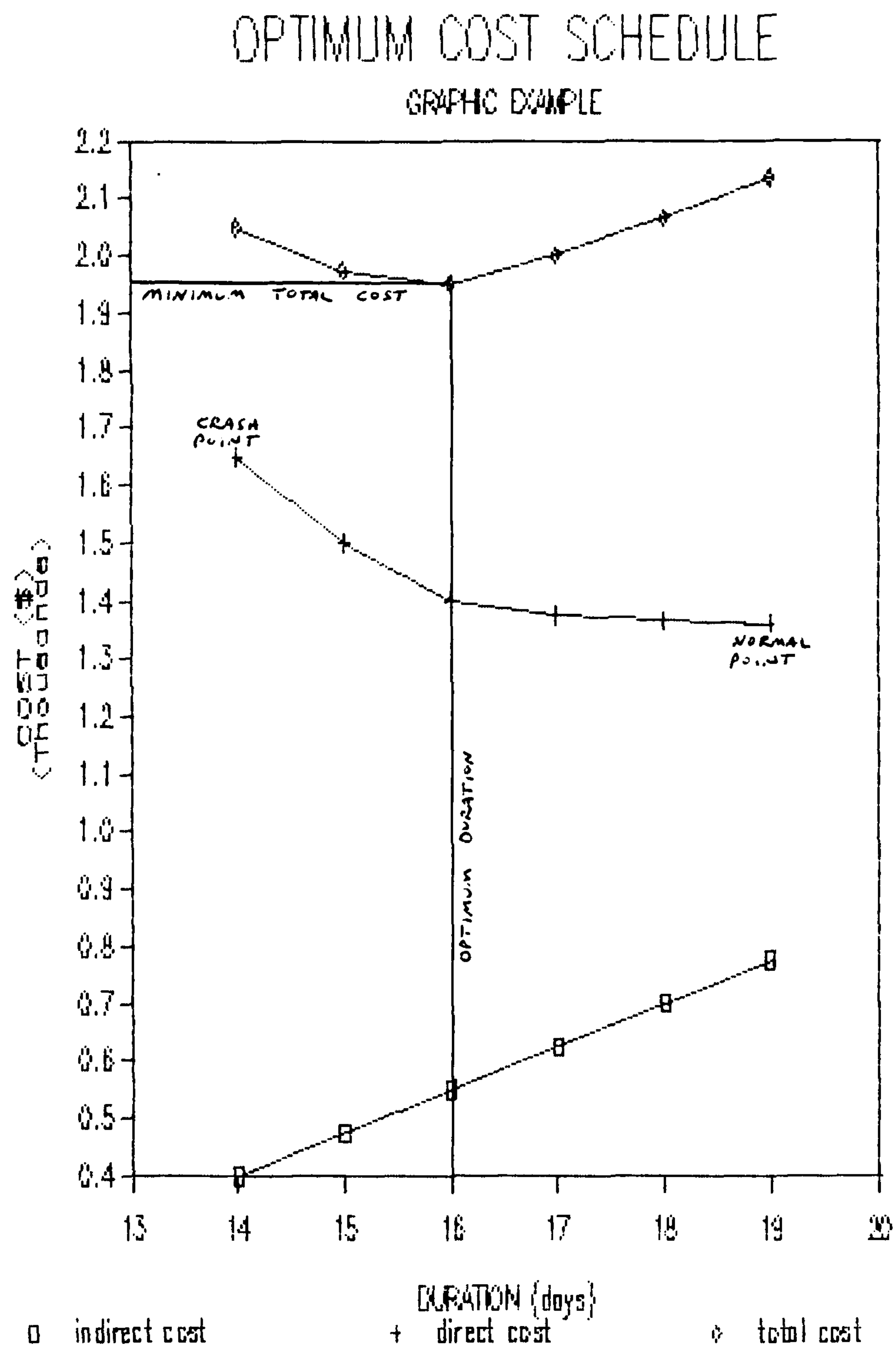
Time Relationship

As described in the previous paragraphs, not all indirect costs are related to the life span of a project. The nature of most job related overheads that do vary with time, however, is the longer the project goes on, the greater the indirect costs become. In general, this increase tends to vary linearly. Likewise, reducing the project duration will decrease indirect costs. This will enable the time-cost trade-off to be of benefit to the scheduler. By combining all the various indirect costs together, one can essentially spread the cost of the project overhead throughout the life of the project at a fixed amount per unit of time. Therefore, the scheduler can calculate exactly how much will be saved in indirect

expenses with any amount of reduction in project duration.

Cost Optimization

Cost optimization scheduling is simply scheduling the project in order to incur the minimum sum of direct and indirect costs. By utilizing the least cost method of reducing project time, the first activities to be compressed are chosen because they increase direct costs the least. Therefore, as project time continues to be reduced, the direct costs of activities to be compressed are steadily increasing. The result is a project direct cost curve that increases as project time is reduced. At some point, this increase in direct costs is greater than the linear decrease in indirect costs. It is, therefore, one time unit prior to this occurrence that produces the project's optimum cost schedule. The duration produced is the project time most economically efficient for the user. In addition to this information, the scheduler also has knowledge of the total cost combinations for other durations. If he is, for some reason, unable to meet the optimum duration, he can determine the best economical solution available to him. Figure 3-2 is a graphic example of an optimum cost schedule. As noted on the graph, the project's optimum duration is 16 days at a total cost of $1950. No other project duration can produce a lesser cost.



OPTIMUM COST SCHEDULE
GRAPHIC EXAMPLE
COST ($) (THOUSANDS)
MINIMUM TOTAL COST
CRASH POINT
NORMAL POINT
OPTIMUM DURATION
DURATION (days)
indirect cost
direct cost
total cost

FIGURE 3-2

CHAPTER IV
PROCEDURAL METHODS OF COST OPTIMIZATION

Manual Methods

There have been procedural techniques proposed for the determination of the project direct cost curve since early in the development of critical path methods. A method for finding an optimal feasible schedule using the primal-dual algorithm for solving linear programs was published in 1961. Other methods utilizing approaches of network flow computations and integer programming were also adapted to solve the problem. All of these methods involved considerable calculation and the use of the digital computer. From the standpoint of the practicing construction manager, none of these methods were suitable. A procedure that could be performed easily with hand calculations was needed. In his 1961 and 1964 publications, John W. Fondahl approached the solution for these direct cost curves in a systematic ordering of hand calculated steps that was designed for the construction manager. (6, p.208) The guidelines for his procedure are still the norm for today, because while the microcomputer is now very common, very few software packages include a computerized solution for this need. As with many new

developments, Fondahl's procedure as originally printed, is not easily understood without his own tabular support documents. For that reason, his procedure will not be quoted step-by-step, but other manual procedures that are more general restatements of his procedure will be highlighted in the following paragraphs.

Procedure Number One

At the end of the last chapter, a total time-cost curve was illustrated. Graphically, it was easy to understand which project duration provided the optimum cost schedule. The following step-by-step manual procedure summarizes the development of this curve from which the optimum project duration is determined:

1. Draw the arrow (or precedence) diagram in the usual manner.
2. Instead of a single estimate of duration, prepare two time estimates as follows:
   a. The normal duration: the time it would take to do the operation in the least expensive manner.
   b. The crash duration: the shortest time in which the operation could possibly be finished, regardless of cost.
3. Estimate the cost of each of these alternatives.
4. Compute and tabulate the cost slope of each operation from the following formula:

$$\text{Cost Slope} = \frac{\text{Crash Cost} - \text{Normal Cost}}{\text{Normal Duration} - \text{Crash Duration}}$$

5. Perform the CPM computations, using the normal durations to determine the normal project duration.
6. Add up the normal costs of all the operations to determine the normal cost of the project. The factors determine the normal point on the project time-cost curve.
7. To find the shape of the project cost curve, shorten the critical operations, one by one, beginning with the operation(s) having the lowest cost slope(s). Each operation is shortened until either:
   a. Its crash time is reached, or

b. A new critical path is formed.

8. When a new critical path is formed, shorten the combination of operations having the lowest combined cost slope. Where several parallel paths exist, it is necessary to shorten simultaneously along each of them if the overall project time is to be reduced.
9. At each step, check to see if float time has been introduced in any of the operations previously shortened. If so, perhaps these operations can be expanded to reduce costs.
10. At each shortening cycle, compute the new project cost and duration. Plot these points on a time-cost graph.
11. Continue until no further shortening is possible. This is the crash point.
12. Compute the indirect project costs and plot them on the same time-cost graph.
13. Add the direct and indirect costs to find the total project cost at each duration.
14. Use the total cost curve to find the optimum time (completion at lowest cost) or the cost of any other desired schedule.

(7, p.131-132)

Procedure Number Two

This procedure utilizes more graphical terms in explaining the expediting process. When using the term diagram, it refers to the CPM network which utilizes the activity on the arrow technique to reflect the project schedule. As will be explained in the next chapter, this is an outdated technique but still used today by more schedulers than any other. This is evident by the fact that this procedure was published in 1986. The process as explained only calculates the direct cost curve. However, one can utilize the last three steps of the previous procedure to calculate the optimum cost schedule. The procedure is as follows:

1. Draw the CPM Diagram
   Show the number of days that each task can be crashed, the cost per day to crash each crashable task, and the total float of each task.
2. Draw Sections
   The sections must pass vertically through the diagram. A section can pass through any task except those that are critical but noncrashable. As the expediting process is continued, additional tasks will become critical, and the remaining crashability of some tasks will be reduced as a result of the expediting process.
3. Find the Least-Cost Section
   The least-cost section is the one in which the sum of the crash costs for the critical tasks that it passed through is the lowest.
4. Crash All the Critical Tasks on the Least-Cost Section
   The number of days these tasks can be crashed will be the least of:
   1. The remaining crashability of any critical task through which the section passes.
   2. The total float of any noncritical task through which the section passes.
5. Reduce the Crashability of Crashed Tasks
   Post the diagram with the remaining crashability of all crashed critical tasks.
6. Reduce the Total Float of Noncritical Tasks Cut by Section
   Reduce the total float of any noncritical task through which the section passed if that noncritical task is concurrent with all crashed critical tasks.
7. Reduce the Total Float of Other Noncritical Tasks
   Reduce the total float of any other noncritical task in the network so that the lowest total float of any arrow originating at a node is equal to the lowest total float of any arrow terminating at the node.
8. Repeat Process as Often as Possible
   Repeat steps 2 through 7 until the project duration has been reduced to the desired value or until it can be reduced no further.

(12, p.306-307)

Procedure Number Three

The following procedure is the one all new Army Engineer lieutenants are presented with at the Engineer Officer Basic Course. The procedure is simplified and requires classroom supervision in order to fully master it.

As with the last procedure, the following procedure only calculates the direct cost curve. To obtain the optimum cost schedule, utilize the last three steps of the first procedure. It is presented as follows:

> 1. Determine the project's normal duration and normal cost; identify those activities which are critical; and mark the critical path(s).
> 2. Establish the time-cost slope for the critical activities.
> 3. Decide which critical activity(ies) should be reduced to effect a project reduction of one time unit. If there is only one critical path, then select the activity with the <u>least</u> (smallest) time-cost slope. When multiple critical paths are present in the CPM network, all combinations of activities which make up the paths must be compared. The technique of selecting the least combination of slopes becomes very complex as the number of paths and critical activities increase. Once the selection is made, the activity(ies) is reduced by one time unit, and a network update is done.
> 4. New critical paths are identified and marked on the updated network. Steps 2 through 4 are repeated until the desired duration is achieved or until all the activities on any <u>one</u> critical path are set to their crash time.
>
> (11, p.35)

<u>Computer Applications</u>

The computer age has certainly arrived. In 1984, it was estimated that 24% of the construction firms in the United States were using computers for scheduling applications. (1, p.4) In the past two years, that figure has most assuredly risen. Today, the microcomputer and related software have become very affordable. Potentially, it provides the contractor an excellent <u>tool</u> for the efficient handling of the vast amount of information that is part of his business. The utilization of this tool with

regards to cost optimization scheduling can only serve to enhance the contractor's decision making process. Such an enhancement can be made by noting Appendix A, which is a listing of a computerized cost optimization scheduling program entitled Crash and written by this report's author. The cost optimization phase of Crash is addressed in great detail in the next chapter. The following paragraphs will focus on the general advantages and limitations of using computer programs for cost optimization scheduling.

Computer Advantages

A greater explanation of the capabilities provided by computer programs in the area of scheduling will be detailed in Chapter V when describing Crash. However, two general points of advantage that the computer provides contractors should now be mentioned. All too often contractors use two arguments to support claims that efforts to make formal plans and schedules are a waste of time. One of the major arguments is the belief that the time and cost of preparing such plans and schedules outweighs the benefits of its use. The other argument that is often mentioned is that given all the uncertain events that characterize a construction project, the prepared plan is soon outdated. The use of the computerized scheduling system can alleviate both of these concerns. The computer makes preparation time a relative nonexistent factor for cost optimization scheduling. Also, the computer can update changes in plans rather effortlessly. Even better,

the computer allows the contractor to perform sensitivity analysis experiments or "what if games" prior to the occurrence of uncertainties. The computer is potentially an outstanding tool that can aid the contractor to better manage his project schedules. (1, p.213)

Program Limitations

Any computer program is based in part on a number of assumptions. A cost optimization scheduling program is no exception. The specific assumptions incorporated into Crash will be addressed in Chapter V. For any program, if the assumptions match the actual characteristics, then the output can be of value. Obviously then, if a program incorporates bad assumptions, it is a liability and provides the user with a false sense of security. Another limitation of the computer program is the inability to make human judgements. For example, a cost optimization program will select the activity that costs the least to shorten, when in fact a human would recognize that this activity may cause a labor dispute or strike and result in greater costs. Also, a machine will not recognize that crashing a particular activity may result in scheduling too many men in too cramped an area. Therefore, as explained earlier, the computer should be valued as a tool enabling the manager to make better decisions, but not without recognizing its limitations.

CHAPTER V
COMPUTERIZED COST OPTIMIZATION WITH CRASH

CPM and Cost Optimization Program Overview

Before a program in cost optimization could be written, there needed to be, as a foundation, a CPM program that would serve as an excellent tool for the scheduling of a project. The best foundation for this would be a program that would be easy to utilize while also providing the essential ingredients to a realistic scheduling model. The following paragraphs provide an overview with regards to the foundation established in the Crash program. This overview will look at capabilities, assumptions, limitations, and the output that is provided by Crash.

Capabilities

For a program to be useful, it must serve its purpose, and it cannot be overwhelming for the operator. Crash was designed with this in mind. The usefulness of a CPM program is extremely dependent on how well it provides the scheduler with a realistic model. The ability of a program not to overwhelm its operator depends to a large extent on how "user friendly" it has been developed. Concerning both the usefulness and user friendly issue is the program's

ability to edit and store data. All of these issues will be addressed in the following sections.

Realistic Model

CPM has proven over the last 30 years to be an excellent tool as a model for the scheduling of construction projects. However, while the ability of the CPM process has improved over those years, many in the industry are still applying the process as it was first developed. The majority of those in the industry still utilize the activity on the arrow technique, as it was first presented. There are two major limitations to this technique that do not enhance the objective of having a realistic model. First is the requirement of providing dummy arrows to show many of the dependency relationships. Second is the inability to show any relationship other than a dependent activity commencing only after the total completion of the preceding activity. Because of these limitations, the activity on the node process of CPM scheduling, or precedence diagram as it is called, was developed. Presently, an increased number in the industry are shifting to this technique. This process, however, only eliminated one of the two limitations from the activity on the arrow technique. No longer is there the requirement of providing dummy arrows in the network. Those in the industry who have made the transition to this process need only to make

a slight improvement to fall in line with the last development of the CPM process. The advanced precedence technique eliminated the inability to show relationships other than starting activities only after preceding activities were completely done. It is my contention that only by utilizing this last development of the CPM process can one in the construction industry truly provide a realistic model to the construction schedule. That is why as the foundation for a program to perform cost optimization scheduling, the basis of the program's CPM calculations must utilize the advanced precedence technique. Crash does, in fact, provide the user with both precedence and advance precedence capabilities. The user of Crash can realistically model overlapping activities, provide for necessary delay time between activities such as with curing time for concrete, or insure other dependency relationships.

User Friendly

While a program must be powerful to be of use to the user, it must also be relatively simple to operate. A first time user should find the program not so difficult that it frightens him away. Rather, it should demonstrate its ease of use. Crash has been developed to provide its users a realistic scheduling model that is easy to utilize. The entire program is run in the form of a type of menu driven, interactive

process. After a set of initial instructions, the operator of the program need only to answer the sequential questions in order to perform any of the various options provided for by Crash. All the questions are answered in a yes or no format except when entering data. The data can be entered in any order without regard to the dependencies of activities. A further ability of the program that helps the user is the option to produce the output on either the screen or paper. Furthermore, the program insures that the output will not scroll across the screen as it is being produced.

Others

Two other capabilities mentioned earlier that affect both the ability of a CPM program to provide a realistic model and to insure ease of use for the operator are the program's ability to file and retrieve data and the editing capabilities of the program. For a network of many activities, it would be very inefficient for the user to repeatedly have to enter data as a new day of work began. Likewise, if a user failed to enter all the data on a project at one time, he would not desire to start back at ground zero when attempting to enter the remaining information. For that reason, any CPM program should have as one of its options the capability to file and retrieve data. Tied into that concept is the ability of a program to

perform editing functions. Inherent in Crash is the means to file, store, and edit data. The editing capabilities of Crash are of two natures. First, as the user enters data one activity at a time, he is asked the question if the entered activity data is correct. If the answer is yes, the program proceeds to the next data entry. If it is no, the user is asked to re-enter the last activity's data. The second phase of editing comes after all the data has been entered. The user is provided a table of recorded data, and if he chooses to alter any piece of information about the project, he can do so. The editing capabilities at this point are: adding activities; deleting activities; changing activity data to include durations, names, costs, dependencies; changing indirect project cost, and changing project headline data. In affect, while this option corrects errors, it also enables the user to run a sensitivity analysis on the schedule. In combination, the storage of data and the editing capability of the Crash program provide the user powerful components to his ability to insure a realistic model and ease of use.

Assumptions

In order to undertake any endeavor, one has to follow some set of rules, principles, or assumptions. To insure that any user can understand the basis of the Crash program, it is best to identify those assumptions on which

it is based. The most important assumption that the program makes is that the user is competent. It assumes that the user has prepared a list of activities with correct dependency relationships and estimates for durations and costs prior to the inputing of data. Another assumption is that the program utilizes the least cost approach to compression of the project. The cost optimization phase of the program assumes a linear relationship for the increased direct cost of crashing an activity. It assumes that the project direct cost curve follows a piece-wise linear curve. The program assumes that all the ingredients that make up the project indirect costs are to be spread out over the project uniformly, resulting in an increasing linear indirect cost curve. Finally, if two or more time durations result in the same project total cost, the program will determine that the cost optimization schedule will be the schedule that is shorter in duration.

Limitations

The Crash program does have some limitations. However, they do not present too great a problem for most project networks. The program is limited to the use of three dependencies for each activity. For most networks, this is sufficient. The program can handle projects of 160 activities or less due to computer memory space available. The output of the bar chart is limited to producing a graph of 440 time units or less. Since the user can select time

units of hours, days, or weeks, this should not limit the use of this feature significantly. These three limitations should not deter any potential user from extensively using Crash.

Graphic and Tabular Output

A CPM arrow diagram, or precedence diagram, can make sense to one who is very knowledgeable about the procedure, but to the common man, it can be extremely confusing, and therefore, of little value to him. For that reason, the Crash program has been developed to provide the user with all of the essential data but in simple formats. There are only two forms of output that the program produces. The two forms are tabular and graphic in nature. There are three different types of tables that are produced and only one kind of graph. An example of these can be found in appendix B. The first document is a table of the recorded project data. Easy to digest, all the data as inputed by the user will appear. The next table is the project's normal completion schedule. It provides the user with each activity's early start, early finish, late start, late finish, and total float times. It also determines critical activities and the project costs and duration. The next output is a bar chart of the schedule. This provides the user an easily understood graphic presentation of the project schedule. If the user is only interested in normal cost scheduling of the project, this will be the total output he will receive concerning the project. However, if

compression or cost optimization is desired, additional reports are produced. As demonstrated in appendix B, the next report is a table of cost iterations demonstrating the changes in project costs as the project is reduced by one time unit until it reaches the project crash point. The following reports produced are similar to the normal cost schedule and bar chart in form but are for the crash point and optimum cost schedules. As demonstrated by the tables and graphs presented in appendix B, the reports provided by Crash are both simple and informative.

Logic of Cost Optimization Phase

By its very nature, cost optimization as demonstrated previously by the manual methods, is a very complicated and arduous task. Developing a program that would accomplish the act of compression would essentially ease the burden on any person required to expedite a schedule. The hard part was to write the computer text that would perform the operation. The logic involved required a program that would work on a potentially infinite sequence of iterative operations producing the combination of compressed activities that would provide the least costly scheduling solution. The following paragraphs will expound on the logic that enables Crash to perform the cost optimization phase.

Determining the Least Costly Reduction

Essential to the calculation of an optimum cost schedule is the ability of Crash to perform calculations

which produce the least costly increase of direct costs for each time unit reduction. In order to produce a program that would be useful and accurate, this key problem had to be solved. The first step in the process is for the program to calculate each critical activity's time-cost slope. After sorting the activities by their respective time-cost slope, with the smallest nonzero value first, the process of determining if compressed activities would reduce project time could commence. The program then begins a series of iterations. To start, the first sorted value is compressed one time unit and the program runs the CPM calculations to see if the project time has also been reduced. If not, the next activity in order is also compressed, and the CPM calculations are run again. This continues until enough activity compressions have been made to reduce the project time by one unit. At this point, it is likely that unnecessary activities have been compressed. Therefore, the costs involved are too high. Thus, the program goes through a series of calculations combining already compressed activities to determine if some less costly combination of those activities will also satisfy a one time unit reduction. When the least cost combination has been determined from those activities, the program checks to see if some other set of unanalyzed activities might prove to be less costly. Eventually, Crash determines the combination of compressed activities which

reduce the project completion by one time unit with the least increase in direct costs.

Determining Optimum Cost

Once Crash has determined the least costly increase of direct costs for a one time unit reduction, the process goes through a series of like repetitions. It continues to reduce the project until every combination of remaining activities that are in themselves compressible cannot compress the project further. The program has at that point determined the project crash completion time. Given the data inputed, the project cannot be completed in any shorter a duration. While that is important information, even more important to the user is the optimum cost at which the project can be completed. Given the assumption that indirect costs are spread out uniformly at a constant cost, the program makes the comparison between that cost and the ever increasing changes in direct costs. The program determines that cost optimization scheduling occurs one time unit prior to the time when increased direct costs are greater than the constant indirect costs. The result is a schedule for the project that establishes the least total cost to the user.

Case Study Example

Perhaps the most efficient manner in which to explain the logic of the cost optimization phase of the Crash program is by way of an example. Throughout the next few pages, an analysis of an example project will be

undertaken. The analysis will utilize a combination of explanations, computer output, and manual calculations based on freehand precedence diagrams. Table 5-1 contains the example's recorded project data. There are no activity descriptions in order to avoid confusion. Simply refer to the activities by their respective activity codes. Table 5-2 is the compression cost iterations output as computed by Crash. As noted by this table, the normal project completion time is 21 days at a total cost of $5880. The crash completion time is 17 days with a total cost of $5750, and the optimum cost completion time is 19 days at the cost of $5720. In order to understand the compression logic used by Crash to produce this table, a step-by-step manual demonstration is provided. Figure 5-1 through figure 5-5 are precedence diagrams of the network at different stages of the compression iterations. Along with the diagrams are discussions of the logic and calculations that the computer makes in deciding which activities to compress.

CASE STUDY EXAMPLE'S PROJECT DATA

THE FOLLOWING DATA IS RECORDED ON THE PROJECT:

PROJECT NAME: CASE STUDY EXAMPLE
SCHEDULED BY: MARK A RONCOLI
DATE ESTIMATED: 19 AUGUST 1986

ACT. CODE	ACTIVITY DESCRIPTION	DURATION NORM/CRASH		ACT. COST NORM/CRASH		DEPENDENCIES
A		2.0		125		-
B		4.0/	3.0	200/	310	-
C		6.0		300		A
D		3.0/	2.0	100/	175	B
F		4.0/	2.0	100/	200	B
E		3.0/	1.0	100/	340	C
H		7.0/	6.0	100/	190	F, D
G		3.0		150		E
I		6.0/	5.0	400/	600	G, H

PROJECT INDIRECT COST= $ 205 PER DAY

TABLE 5-1

CASE STUDY EXAMPLE'S COMPRESSION ITERATIONS

COMPRESSION COST ITERATIONS

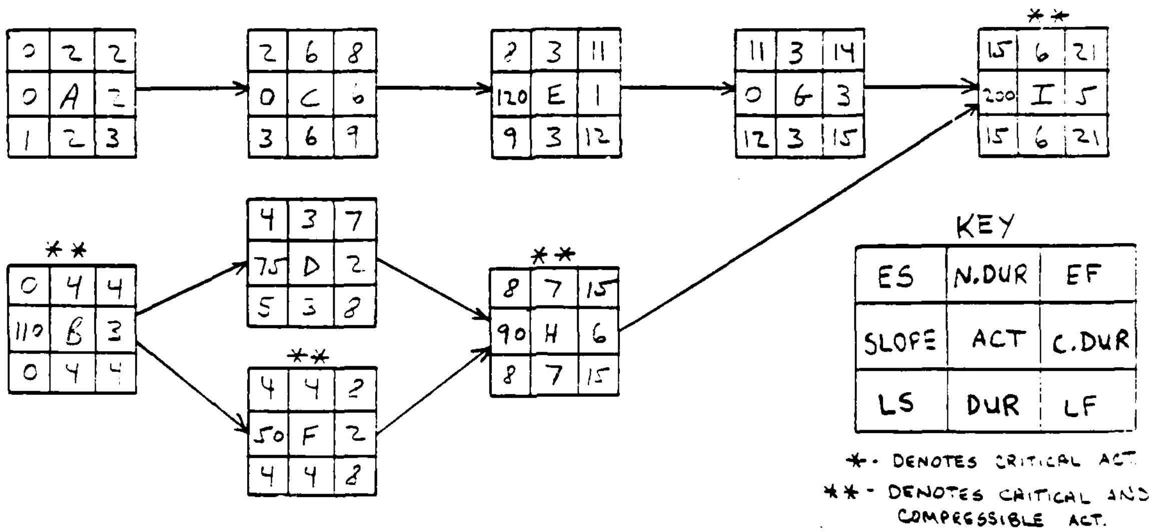
NORMAL PROJECT DURATION: 21 | NORMAL PROJECT DIRECT COST: $ 1575
NORMAL PROJECT INDIRECT COST: $ 4305 | NORMAL PROJECT TOTAL COST: $ 5880

COMPRESSED ACTIVITY(S)	INCREASED DIRECT COSTS	PROJECT DIRECT COSTS	PROJECT INDIRECT CT	PROJECT TOTAL COST	PROJECT DURATION
F	50	1625	4100	5725	20
I	200	1825	3895	5720	19
H, E	210	2035	3690	5725	18
B, E	230	2265	3485	5750	17

TABLE 5-2

Example Project Precedence Diagram

Compressing from 21 to 20 days



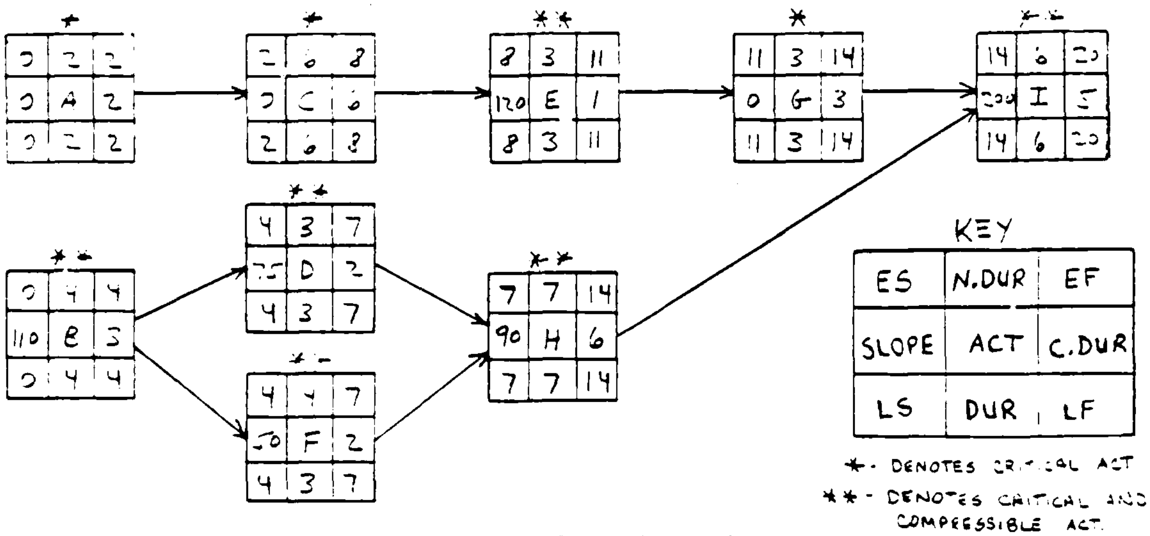
Critical Activities	Time-Cost Slope
F	50
H	90
B	110
I	200

As noted by the diagram above, there are four activities that are critical at a project duration of 21 days. These are activities B, F, H, and I. All four of these activities are compressible. Crash will sort these four activities by the least time-cost slope and select activity F to be the first trial compression iteration. Compressing activity F one day at an increased cost of $50 will reduce the project time to 20 days.

Figure 5-1

Example Project Precedence Diagram

Compression from 20 to 19 days



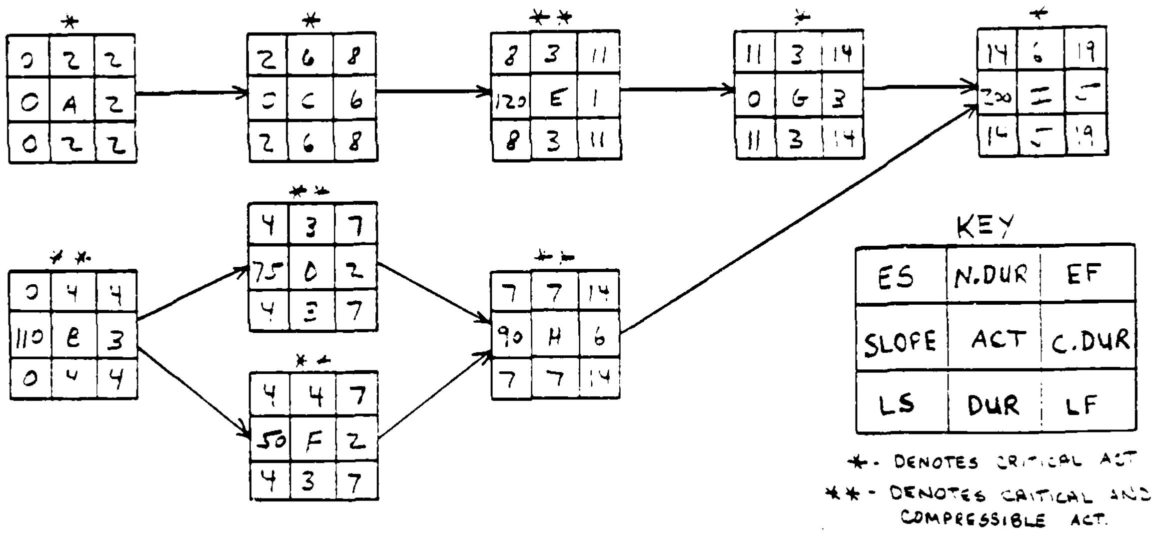
Compressible Activities	Time-Cost Slope
F	50
D	75
H	90
B	110
E	120
I	200

As noted above, all the project activities are now critical. However, activities A, C, and G are not compressible. The sorting of the compressible activities by the least time-cost slope is shown above. To optimize the schedule, Crash will proceed through a series of logic phases. The program will first compress each of these activities one day, in the sorted order, until the project is reduced one day. In this case, the last activity the program compressed that resulted in the project reduction was E. Not all of the activities needed to be compressed. Therefore, the program determines which combination of activities, to include activity E, provides the project time reduction at the least increased direct cost. In this case, activities H and E, at an increased cost of $210, are chosen. The program then checks to see if there are some other activities whose time-cost slopes are greater than E's but less than the already determined cost of $210. These activities would then be examined to see if they could be combined in a manner that would provide a cheaper solution. In this case, the program would choose to compress activity I at an increased cost of $200, as the final solution to reducing the project from 20 to 19 days.

Figure 5-2

Example Project Precedence Diagram

Compression from 19 to 18 days



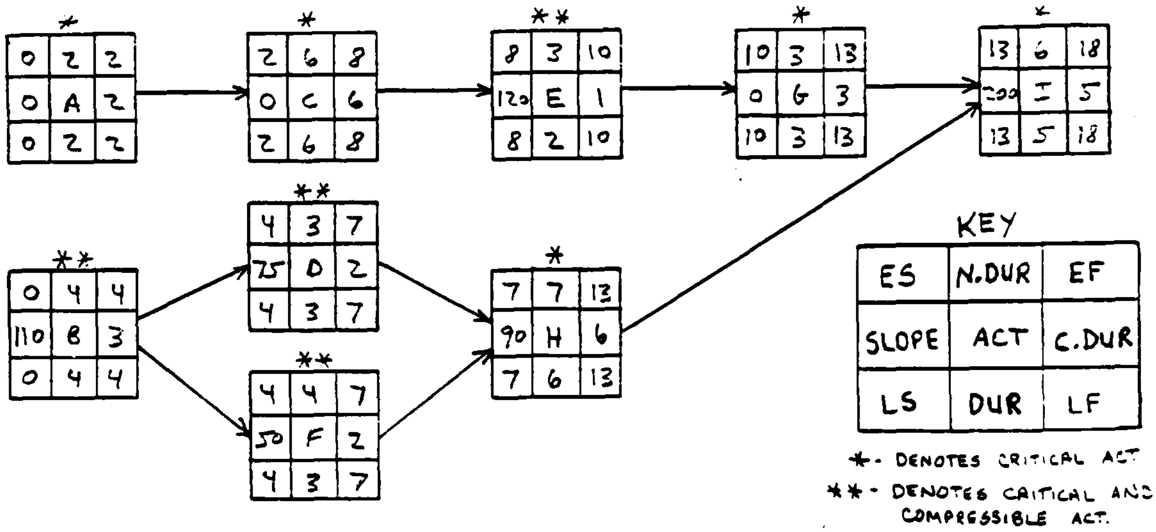
Compressible Activities	Time-Cost Slope
F	50
D	75
H	90
B	110
E	120

All the activities continue to be critical and will remain so throughout the project compression iterations. In addition to activities A, C, and G, activity I is now also not compressible. The program will utilize the same logic described on the previous page and determine that compressing activities H and E, at a cost of $210, will compress the project one day at the least increase of direct costs.

Figure 5-3

Example Project Precedence Diagram

Compression from 18 to 17 days



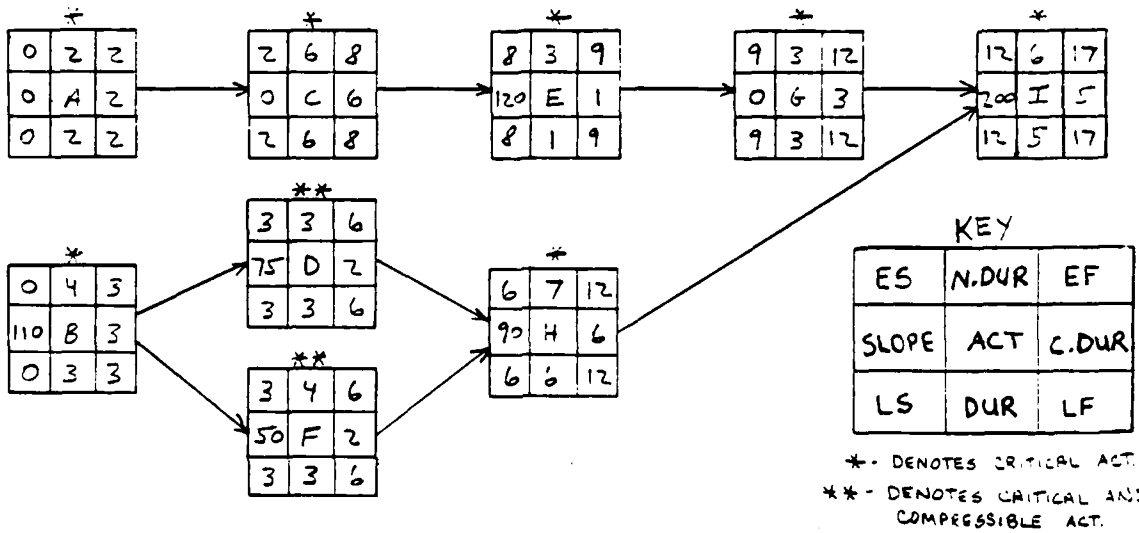
Compressible Activities	Time-Cost Slope
F	50
D	75
B	110
E	120

Because of the last compression iteration, activity H is now also not compressible. Again, the program determines that activity E is the last activity to be compressed which results in a one day project reduction. As before, not all the activities compressed needed to be. The process by which the program determines which combination with activity E will provide the least increase in costs follows: First, the program will add back one day to each of the compressed activities except activity E. The program then calculates that a compression of activity E alone will not reduce the project duration. The activities are again compressed one by one in the sorted order until the project duration shows a reduction. In this case, activity D is the last to be compressed, and in combination with activities E, and F, they will reduce the project at $245. Next, Crash will check to see if there are other activities whose time-cost slopes are greater than D's, but in combination with E's, less than the $245. In this case, the program would choose to compress activities E and B at an increased cost of $230. As described, this process is cyclical in nature and can involve numerous iterations for larger projects.

Figure 5-4

Example Project Precedence Diagram

Compression limited to 17 days



Compressible Activities	Time-Cost Slope
F	50
D	75

The last compression iteration resulted in activities F and D being the only activities remaining compressible. Crash will determine that even if both activities are compressed, it will not result in any further reduction in project duration. Therefore, those two activities are not reduced. The project's crash point is 17 days.

Figure 5-5

As demonstrated by the notations to the preceding precedence diagrams, the program's logic for determining which activities to compress is quite involved. It is that fact which made the development of Crash quite a challenge.

Benefits of Using Crash

The benefits of utilizing the Crash program are many. Along with so many computer programs of today's information era, Crash provides an accurate means of information without the requirement of intensive manpower. What makes this program so very special is its numerous capabilities. By providing a realistic CPM scheduling capability while insuring a "user friendly" approach, almost anyone requiring to schedule a project should be fully equipped to do so. Freeing manpower to concentrate on other areas of interest instead of having to make manual compression calculations, while insuring error free results, provides the user a definite advantage. The production of informative and professional tables and graphs enables those in the construction industry to make intelligent decisions concerning scheduling strategies. Yes, cost optimization is of great concern to those trying to make it in the construction business, and Crash can make it a lot easier to understand.

CHAPTER VI
CONCLUSION

Summary

In the preceding chapters, the concepts regarding cost optimization scheduling and a computerized solution to this process were presented. The reasons for expediting a project and the various approaches and strategies for doing so were described. The method utilized by this report to reduce project time was to compress the critical activities which increased direct costs the least. As a result of a shorter project duration, indirect costs decreased. The increase of direct costs was then compared with the decrease in indirect costs, to produce the optimum cost schedule. The time savings provided by a computer program in performing the numerous calculations was also highlighted. The assumptions required to utilize this procedure, both manually and with the computer, are the same and were described. Finally, a computer program written by the author and called Crash was presented.

Assertion

Given the assumption that a contractor wishes to successfully manage his business, this report has described one aspect of his managerial role. The contractor's

ability to determine the optimum cost schedule for a project relates directly to his firm's profitability. A computer software program for determining the optimum cost schedule is vital to the practicality of utilizing this management tool. The computer program Crash has been presented to provide the contractor with this tool. Now, there is no excuse for anyone failing to perform cost optimization calculations.

APPENDIX A
LISTING OF CRASH PROGRAM

```
10 REM
20 REM     PROGRAM INFORMATION
30 REM
40 KEY OFF
50 CLS
60 PRINT "CPM & COST OPTIMIZATION SCHEDULING"
70 PRINT "PREPARED BY MARK A RONCOLI, US ARMY CAPTAIN
80 PRINT "IN 1986 WHILE A GRADUATE STUDENT AT THE UNIVERSITY OF FLORIDA"
90 PRINT " "
100 PRINT " "
110 PRINT "THIS SCHEDULING PROGRAM ALLOWS YOU TO USE EITHER THE PRECEDENCE METHO
D"
120 PRINT "OR ADVANCED PRECEDENCE METHOD OF CPM"
130 PRINT " "
140 PRINT "TO UTILIZE THE PROGRAM USE (ALL CAPS)"
150 PRINT "ALSO YOU WILL NEED THE ACTIVITY LIST FOR THE PROJECT: (WHICH INCLUDES
)"
160 PRINT "ACTIVITY CODES/ ACTIVITY DESCRIPTIONS/ DURATIONS/ DEPENDENCIES"
170 PRINT "AND CRASH TIMES AND COSTS IF COST OPTIMIZATION IS DESIRED"
180 PRINT " "
190 PRINT "THE PROGRAM HAS THE FOLLOWING LIMITATIONS:"
200 PRINT "IT WILL ONLY HANDLE UP TO THREE DEPENDENCIES"
210 PRINT "THE BAR CHART WHILE FLEXIBLE WILL ONLY DISPLAY UP TO 440 TIME UNITS"
220 PRINT "THE PROGRAM IS LIMITED TO 160 ACTIVITIES DUE TO COMPUTER MEMORY SPACE
"
230 PRINT " "
240 PRINT " "
250 INPUT "TO START THE PROGRAM HIT THE ENTER KEY";VV$
260 DIM J$(4)
270 DIM A$(160),AD$(160),B$(160),C$(160),D$(160),D(160),ES(160),EF(160),LS(160),
LF(160)
280 DIM LLF(160),LLS(160),CP$(160),F(160),BSS(160),CSS(160),DSS(160),EES(160)
290 DIM BSF(160),CSF(160),DSF(160),BFF(160),CFF(160),DFF(160),CD(160),XD(160)
300 DIM NNC(160),NCC(160),TCS(160),IJ(160),JI(160),CCD(160),NCT(160)
310 DIM HH(160),NPT(160),DPT(160),JKL(160),EE(160)
320 CLS
330 INPUT "DO YOU WISH TO RECALL AN ESTABLISHED PROJECT DATA FILE (Y OR N)";WT$
340 IF WT$<>"Y" GOTO 410
350 INPUT "WHICH PROJECT DO YOU WISH TO RECALL";F$
360 GOSUB 9450
370 GOTO 4260
380 REM
390 REM     ENTER DATA
400 REM
410 CLS
420 PRINT "ENTER NEW PROJECT DATA"
430 INPUT "ENTER THE NAME OF WHO IS SCHEDULING THE PROJECT";J$(2)
440 INPUT "ENTER TODAY'S DATE";J$(3)
450 INPUT "ENTER PROJECT NAME";J$(1)
460 INPUT "ENTER NUMBER OF ACTIVITIES";N
470 INPUT "ENTER UNIT OF TIME. 1=HOURS  2=DAYS  3=WEEKS";J$(4)
480 INPUT "WILL YOU REQUIRE ADVANCE PRECEDENCE CAPABILITIES (Y OR N)";KK$
490 INPUT "DO YOU WANT TO ENTER ACTIVITY DIRECT COST DATA (Y OR N)";ACTDC$
500 IF ACTDC$<>"Y" GOTO 560
```

```
510 INPUT "WILL YOU REQUIRE COMPRESSION CAPABILITIES (Y OR N)";PP$
520 IF PP$<>"Y" GOTO 560
530 INPUT "DO YOU WANT TO ENTER PROJECT INDIRECT COST DATA (Y OR N)";ICD$
540 IF ICD$<>"Y" GOTO 560
550 INPUT "ENTER THE INDIRECT COST PER UNIT OF TIME";INDCOST
560 CLS
570 PRINT "BEGIN ENTERING ACTIVITY DATA"
580 IF PP$<>"Y" GOTO 620
590 PRINT " "
600 PRINT "NOTE:  IF AN ACTIVITY CANNOT BE CRASHED, THEN IF ASKED TO ENTER THE C
RASH TIME"
610 PRINT "AND COST, ENTER IT EXACTILY AS YOU ENTERED TO NORMAL TIME AND COST."
620 PRINT " "
630 FOR I=1 TO N
640 INPUT "ENTER ACTIVITY CODE";A$(I)
650 INPUT "ENTER ACTIVITY DESCRIPTION";AD$(I)
660 IF PP$<>"Y" GOTO 720
670 INPUT "ENTER THE NORMAL TIME DURATION";D(I)
680 INPUT "ENTER THE CRASH TIME DURATION";CD(I)
690 INPUT "ENTER THE NORMAL COST";NNC(I)
700 INPUT "ENTER THE CRASH COST";NCC(I)
710 GOTO 770
720 INPUT "ENTER DURATION";D(I)
730 CD(I)=D(I)
740 IF ACTDC$<>"Y" GOTO 770
750 INPUT "ENTER COST";NNC(I)
760 NCC(I)=NNC(I)
770 PRINT "YOU MAY ENTER UP TO THREE PRECEDENT ACTIVITIES FOR EACH ACTIVITY"
780 INPUT "ENTER DEPENDENCY 1.  IF NONE, ENTER: 0";B$(I)
790 IF B$(I)="0" GOTO 1080
800 IF KK$<>"Y" GOTO 880
810 INPUT "IF THE DEPENDENCY IS NORMAL ENTER: 0.  ELSE ENTER: SF, SS, OR FF";AA$

820 IF AA$<>"SF" GOTO 840
830 INPUT "ENTER THE START-TO-FINISH TIME";BSF(I)
840 IF AA$<>"SS" GOTO 860
850 INPUT "ENTER THE START-TO-START TIME";BSS(I)
860 IF AA$<>"FF" GOTO 880
870 INPUT "ENTER THE FINISH-TO-FINISH TIME";BFF(I)
880 INPUT "ENTER DEPENDENCY 2.  IF NONE, ENTER: 0";C$(I)
890 IF C$(I)="0" GOTO 1080
900 IF KK$<>"Y" GOTO 980
910 INPUT "IF THE DEPENDENCY IS NORMAL ENTER: 0.  ELSE ENTER: SF, SS, OR FF";AA$

920 IF AA$<>"SF" GOTO 940
930 INPUT "ENTER THE START-TO-FINISH TIME";CSF(I)
940 IF AA$<>"SS" GOTO 960
950 INPUT "ENTER THE START-TO-START TIME";CSS(I)
960 IF AA$<>"FF" GOTO 980
970 INPUT "ENTER THE FINISH-TO-FINISH TIME";CFF(I)
980 INPUT "ENTER DEPENDENCY 3.  IF NONE, ENTER: 0";D$(I)
990 IF D$(I)="0" GOTO 1080
1000 IF KK$<>"Y" GOTO 1080
```

```
1010 INPUT "IF THE DEPENDENCY IS NORMAL ENTER: O.  ELSE ENTER: SF, SS, OR FF";AA
$
1020 IF AA$<>"SF" GOTO 1040
1030 INPUT "ENTER THE START-TO-FINISH TIME";DSF(I)
1040 IF AA$<>"SS" GOTO 1060
1050 INPUT "ENTER THE START-TO-START TIME";DSS(I)
1060 IF AA$<>"FF" GOTO 1080
1070 INPUT "ENTER THE FINISH-TO-FINISH TIME";DFF(I)
1080 INPUT "IS THE DATA CORRECT? Y OR N";Z$
1090 IF Z$="Y" GOTO 1120
1100 PRINT "RE-ENTER ALL OF THE DATA FOR THE LAST ACTIVITY"
1110 GOTO 640
1120 CLS
1130 NEXT I
1140 REM
1150 REM      SORT BY PRECEDENCE
1160 REM
1170 FOR NM=1 TO N
1180 FOR I=NM+1 TO N
1190 IF B$(NM)=A$(I) GOTO 1240
1200 IF C$(NM)=A$(I) GOTO 1240
1210 IF D$(NM)=A$(I) GOTO 1240
1220 NEXT I
1230 GOTO 1830
1240 AA$=A$(I)
1250 ADAD$=AD$(I)
1260 DD=D(I)
1270 BB$=B$(I)
1280 CC$=C$(I)
1290 DD$=D$(I)
1300 BSSS=BSS(I)
1310 CSSS=CSS(I)
1320 DSSS=DSS(I)
1330 BSFF=BSF(I)
1340 CSFF=CSF(I)
1350 DSFF=DSF(I)
1360 BFFF=BFF(I)
1370 CFFF=CFF(I)
1380 DFFF=DFF(I)
1390 CD1=CD(I)
1400 NNC1=NNC(I)
1410 NCC1=NCC(I)
1420 XD1=XD(I)
1430 TCS1=TCS(I)
1440 IJ1=IJ(I)
1450 JI1=JI(I)
1460 CCD1=CCD(I)
1470 NCT1=NCT(I)
1480 NPT1=NPT(I)
1490 DPT1=DPT(I)
1500 HH1=HH(I)
1510 JKL1=JKL(I)
```

```
1520 EE1=EE(I)
1530 EE(I)=EE(NM):EE(NM)=EE1
1540 A$(I)=A$(NM):A$(NM)=AA$
1550 AD$(I)=AD$(NM):AD$(NM)=ADAD$
1560 D(I)=D(NM):D(NM)=DD
1570 B$(I)=B$(NM):B$(NM)=BB$
1580 C$(I)=C$(NM):C$(NM)=CC$
1590 D$(I)=D$(NM):D$(NM)=DD$
1600 BSS(I)=BSS(NM):BSS(NM)=BSSS
1610 CSS(I)=CSS(NM):CSS(NM)=CSSS
1620 DSS(I)=DSS(NM):DSS(NM)=DSSS
1630 BSF(I)=BSF(NM):BSF(NM)=BSFF
1640 CSF(I)=CSF(NM):CSF(NM)=CSFF
1650 DSF(I)=DSF(NM):DSF(NM)=DSFF
1660 BFF(I)=BFF(NM):BFF(NM)=BFFF
1670 CFF(I)=CFF(NM):CFF(NM)=CFFF
1680 DFF(I)=DFF(NM):DFF(NM)=DFFF
1690 CD(I)=CD(NM):CD(NM)=CD1
1700 NNC(I)=NNC(NM):NNC(NM)=NNC1
1710 NCC(I)=NCC(NM):NCC(NM)=NCC1
1720 XD(I)=XD(NM):XD(NM)=XD1
1730 TCS(I)=TCS(NM):TCS(NM)=TCS1
1740 IJ(I)=IJ(NM):IJ(NM)=IJ1
1750 JI(I)=JI(NM):JI(NM)=JI1
1760 CCD(I)=CCD(NM):CCD(NM)=CCD1
1770 NCT(I)=NCT(NM):NCT(NM)=NCT1
1780 NPT(I)=NPT(NM):NPT(NM)=NPT1
1790 DPT(I)=DPT(NM):DPT(NM)=DPT1
1800 HH(I)=HH(NM):HH(NM)=HH1
1810 JKL(I)=JKL(NM):JKL(NM)=JKL1
1820 NM=NM-1
1830 NEXT NM
1840 REM
1850 REM      DIRECT COST CALCULATIONS
1860 REM
1870 IF VP>0 GOTO 1900
1880 CHANGE=DIRC
1890 VP=VP+1
1900 DCOST=0
1910 FOR I=1 TO N
1920 DCOST=DCOST+NNC(I)
1930 NEXT I
1940 DIRC=DCOST
1950 INCR=DIRC-CHANGE
1960 REM
1970 REM      ES & EF CALCULATIONS
1980 REM
1990 FOR I=1 TO N
2000 IF B$(I)="O" THEN ES(I)=0:EF(I)=D(I)
2010 NEXT I
2020 PP=0
2030 FOR I=1 TO N
```

```
2040 FOR J=1 TO N
2050 IF A$(J)=B$(I) OR A$(J)=C$(I) OR A$(J)=D$(I) THEN 2110
2060 NEXT J
2070 EF(I)=ES(I)+D(I)
2080 PP=0
2090 NEXT I
2100 GOTO 2170
2110 IF PP<EF(J) THEN PP=EF(J)
2120 ES(I)=PP
2130 GOTO 2060
2140 REM
2150 REM     RECALCULATING FOR SS, SF, AND FF TIME FACTORS
2160 REM
2170 FOR I=1 TO N
2180 IF B$(I)="0" GOTO 2690
2190 IF BSS(I)=0 AND BSF(I)=0 AND BFF(I)=0 GOTO 2290
2200 FOR QQ=1 TO I-1
2210 IF A$(QQ)=B$(I) GOTO 2230
2220 NEXT QQ
2230 IF BSS(I)<>0 THEN EES(I)=ES(QQ)+BSS(I) ELSE 2250
2240 GOTO 2330
2250 IF BSF(I)<>0 THEN EES(I)=EF(QQ)+BSF(I) ELSE 2270
2260 GOTO 2330
2270 EES(I)=EF(QQ)+BFF(I)-D(I)
2280 GOTO 2330
2290 FOR QQ=1 TO I-1
2300 IF A$(QQ)=B$(I) GOTO 2320
2310 NEXT QQ
2320 EES(I)=ES(QQ)+D(QQ)
2330 IF C$(I)="0" GOTO 2670
2340 IF CSS(I)<>0 OR CSF(I)<>0 OR CFF(I)<>0 GOTO 2400
2350 FOR QQ=1 TO I-1
2360 IF A$(QQ)=C$(I) GOTO 2380
2370 NEXT QQ
2380 IF EES(I)<EF(QQ) THEN EES(I)=EF(QQ)
2390 GOTO 2500
2400 FOR QQ=1 TO I-1
2410 IF A$(QQ)=C$(I) GOTO 2430
2420 NEXT QQ
2430 IF CSS(I)=0 GOTO 2460
2440 IF EES(I)<ES(QQ)+CSS(I) THEN EES(I)=ES(QQ)+CSS(I)
2450 GOTO 2500
2460 IF CSF(I)=0 GOTO 2490
2470 IF EES(I)<EF(QQ)+CSF(I) THEN EES(I)=EF(QQ)+CSF(I)
2480 GOTO 2500
2490 IF EES(I)<EF(QQ)+CFF(I)-D(I) THEN EES(I)=EF(QQ)+CFF(I)-D(I)
2500 IF D$(I)="0" GOTO 2670
2510 IF DSS(I)<>0 OR DSF(I)<>0 OR DFF(I)<>0 GOTO 2570
2520 FOR QQ=1 TO I-1
2530 IF A$(QQ)=D$(I) GOTO 2550
2540 NEXT QQ
2550 IF EES(I)<EF(QQ) THEN EES(I)=EF(QQ)
```

```
2560 GOTO 2670
2570 FOR QQ=1 TO I-1
2580 IF A$(QQ)=D$(I) GOTO 2600
2590 NEXT QQ
2600 IF DSS(I)=0 GOTO 2630
2610 IF EES(I)<ES(QQ)+DSS(I) THEN EES(I)=ES(QQ)+DSS(I)
2620 GOTO 2670
2630 IF DSF(I)=0 GOTO 2660
2640 IF EES(I)<EF(QQ)+DSF(I) THEN EES(I)=EF(QQ)+DSF(I)
2650 GOTO 2670
2660 IF EES(I)<EF(QQ)+DFF(I)-D(I) THEN EES(I)=EF(QQ)+DFF(I)-D(I)
2670 ES(I)=EES(I)
2680 EF(I)=ES(I)+D(I)
2690 NEXT I
2700 REM
2710 REM      LS & LF CALCULATIONS
2720 REM
2730 TT=0
2740 FOR I=1 TO N
2750 IF EF(I)>TT THEN TT=EF(I)
2760 NEXT I
2770 FOR I=1 TO N
2780 LF(I)=TT
2790 LS(I)=LF(I)-D(I)
2800 NEXT I
2810 FOR I=N TO 1 STEP -1
2820 FOR G=N TO 1 STEP -1
2830 IF A$(I)<>B$(G) GOTO 2960
2840 IF BSS(G)=0 AND BSF(G)=0 AND BFF(G)=0 GOTO 3220
2850 IF BSS(G)=0 GOTO 2890
2860 LLF(I)=LS(G)+D(I)-BSS(G)
2870 LLS(I)=LLF(I)-D(I)
2880 GOTO 3240
2890 IF BSF(G)=0 GOTO 2930
2900 LLF(I)=LS(G)-BSF(G)
2910 LLS(I)=LLF(I)-D(I)
2920 GOTO 3240
2930 LLF(I)=LF(G)-BFF(G)
2940 LLS(I)=LLF(I)-D(I)
2950 GOTO 3240
2960 IF A$(I)<>C$(G) GOTO 3090
2970 IF CSS(G)=0 AND CSF(I)=0 AND CFF(I)=0 GOTO 3220
2980 IF CSS(G)=0 GOTO 3020
2990 LLF(I)=LS(G)+D(I)-CSS(G)
3000 LLS(I)=LLF(I)-D(I)
3010 GOTO 3240
3020 IF CSF(G)=0 GOTO 3060
3030 LLF(I)=LS(G)-CSF(G)
3040 LLS(I)=LLF(I)-D(I)
3050 GOTO 3240
3060 LLF(I)=LF(G)-CFF(G)
3070 LLS(I)=LLF(I)-D(I)
```

```
3080 GOTO 3240
3090 IF A$(I)<>D$(G) GOTO 3260
3100 IF DSS(G)=0 AND DSF(I)=0 AND DFF(I)=0 GOTO 3220
3110 IF DSS(G)=0 GOTO 3150
3120 LLF(I)=LS(G)+D(I)-DSS(G)
3130 LLS(I)=LLF(I)-D(I)
3140 GOTO 3240
3150 IF DSF(G)=0 GOTO 3190
3160 LLF(I)=LS(G)-DSF(G)
3170 LLS(I)=LLF(I)-D(I)
3180 GOTO 3240
3190 LLF(I)=LF(G)-DFF(G)
3200 LLS(I)=LLF(I)-D(I)
3210 GOTO 3240
3220 LLF(I)=LS(G)
3230 LLS(I)=LLF(I)-D(I)
3240 IF LLF(I)<LF(I) THEN LF(I)=LLF(I)
3250 IF LLS(I)<LS(I) THEN LS(I)=LLS(I)
3260 NEXT G
3270 NEXT I
3280 FOR I=1 TO N
3290 F(I)=LF(I)-EF(I)
3300 IF F(I)=0 GOTO 3340
3310 CP$(I)="-"
3320 NEXT I
3330 GOTO 3390
3340 CP$(I)="C"
3350 GOTO 3320
3360 REM
3370 REM      CALCULATING PROJECT INDIRECT COST
3380 REM
3390 INDCT=INDCOST*TT
3400 REM
3410 REM      SORT BY EARLY START
3420 REM
3430 FOR NM=1 TO N
3440 FOR I=NM+1 TO N
3450 IF ES(NM)>ES(I) GOTO 3470
3460 GOTO 4170
3470 AA$=A$(I)
3480 ADAD$=AD$(I)
3490 DD=D(I)
3500 BB$=B$(I)
3510 CC$=C$(I)
3520 DD$=D$(I)
3530 CD1=CD(I)
3540 NNC1=NNC(I)
3550 NCC1=NCC(I)
3560 XD1=XD(I)
3570 TCS1=TCS(I)
3580 IJ1=IJ(I)
3590 JI1=JI(I)
```

```
3600 CCD1=CCD(I)
3610 NCT1=NCT(I)
3620 NPT1=NPT(I)
3630 DPT1=DPT(I)
3640 HH1=HH(I)
3650 JKL1=JKL(I)
3660 BSSS=BSS(I)
3670 CSSS=CSS(I)
3680 DSSS=DSS(I)
3690 BSFF=BSF(I)
3700 CSFF=CSF(I)
3710 DSFF=DSF(I)
3720 BFFF=BFF(I)
3730 CFFF=CFF(I)
3740 DFFF=DFF(I)
3750 EE1=EE(I)
3760 EE(I)=EE(NM):EE(NM)=EE1
3770 A$(I)=A$(NM):A$(NM)=AA$
3780 AD$(I)=AD$(NM):AD$(NM)=ADAD$
3790 D(I)=D(NM):D(NM)=DD
3800 B$(I)=B$(NM):B$(NM)=BB$
3810 C$(I)=C$(NM):C$(NM)=CC$
3820 D$(I)=D$(NM):D$(NM)=DD$
3830 CD(I)=CD(NM):CD(NM)=CD1
3840 NNC(I)=NNC(NM):NNC(NM)=NNC1
3850 NCC(I)=NCC(NM):NCC(NM)=NCC1
3860 XD(I)=XD(NM):XD(NM)=XD1
3870 TCS(I)=TCS(NM):TCS(NM)=TCS1
3880 IJ(I)=IJ(NM):IJ(NM)=IJ1
3890 JI(I)=JI(NM):JI(NM)=JI1
3900 CCD(I)=CCD(NM):CCD(NM)=CCD1
3910 NCT(I)=NCT(NM):NCT(NM)=NCT1
3920 NPT(I)=NPT(NM):NPT(NM)=NPT1
3930 DPT(I)=DPT(NM):DPT(NM)=DPT1
3940 HH(I)=HH(NM):HH(NM)=HH1
3950 JKL(I)=JKL(NM):JKL(NM)=JKL1
3960 BSS(I)=BSS(NM):BSS(NM)=BSSS
3970 CSS(I)=CSS(NM):CSS(NM)=CSSS
3980 DSS(I)=DSS(NM):DSS(NM)=DSSS
3990 BSF(I)=BSF(NM):BSF(NM)=BSFF
4000 CSF(I)=CSF(NM):CSF(NM)=CSFF
4010 DSF(I)=DSF(NM):DSF(NM)=DSFF
4020 BFF(I)=BFF(NM):BFF(NM)=BFFF
4030 CFF(I)=CFF(NM):CFF(NM)=CFFF
4040 DFF(I)=DFF(NM):DFF(NM)=DFFF
4050 ESES=ES(I)
4060 EFEF=EF(I)
4070 LSLS=LS(I)
4080 LFLF=LF(I)
4090 CPCP$=CP$(I)
4100 FF=F(I)
4110 ES(I)=ES(NM):ES(NM)=ESES
```

```
4120 EF(I)=EF(NM):EF(NM)=EFEF
4130 LS(I)=LS(NM):LS(NM)=LSLS
4140 LF(I)=LF(NM):LF(NM)=LFLF
4150 CP$(I)=CP$(NM):CP$(NM)=CPCP$
4160 F(I)=F(NM):F(NM)=FF
4170 NEXT I
4180 NEXT NM
4190 IF OPTCOST=1 GOTO 5470
4200 IF GH=1 GOTO 5470
4210 IF SP>0 GOTO 13550
4220 IF YB>0 GOTO 9840
4230 REM
4240 REM     PRINTING PROJECT DATA
4250 REM
4260 INDCT=INDCOST*TT
4270 IF INDCOST<>0 THEN ICD$="Y" ELSE ICD$="N"
4280 YY$="N"
4290 DEVICE$="SCRN:":WIDTH "SCRN:",80
4300 GOTO 4320
4310 DEVICE$="LPT1:":WIDTH "LPT1:",80
4320 OPEN DEVICE$ FOR OUTPUT AS #1
4330 PRINT #1, TAB(17);"THE FOLLOWING DATA IS RECORDED ON THE PROJECT:"
4340 PRINT #1, " "
4350 PRINT #1, " "
4360 PRINT #1, " "
4370 PRINT #1, TAB(28);"PROJECT NAME:   ";J$(1)
4380 PRINT #1, TAB(28);"SCHEDULED BY:   ";J$(2)
4390 PRINT #1, TAB(28);"DATE ESTIMATED: ";J$(3)
4400 PRINT #1, " "
4410 PRINT #1, " "
4420 PRINT #1, TAB(6);"ACT.";TAB(37);"DURATION";TAB(50);"ACT. COST"
4430 PRINT #1, TAB(6);"CODE";TAB(13);"ACTIVITY DESCRIPTION";TAB(36);"NORM/CRASH"
;TAB(49);"NORM/CRASH";TAB(63);"DEPENDENCIES"
4440 PRINT #1, TAB(6);"----";TAB(13);"--------------------";TAB(36);"------
;TAB(49);"----------";TAB(63);"------------"
4450 LL=0
4460 FOR I=1 TO N
4470 IF YY$="Y" GOTO 4520
4480 LL=LL+1
4490 IF LL<>15 GOTO 4520
4500 LL=0
4510 INPUT "HIT THE ENTER KEY TO CONTINUE";WW$
4520 F1$="###.#"
4530 F2$="##"
4540 F3$="#####"
4550 PRINT #1, TAB(6);A$(I);TAB(16);AD$(I);TAB(35);
4560 PRINT #1, USING F1$;D(I);
4570 IF D(I)=CD(I) GOTO 4600
4580 PRINT #1, "/";
4590 PRINT #1, USING F1$;CD(I);
4600 PRINT #1, TAB(48);
4610 PRINT #1, USING F3$;NNC(I);
4620 IF NNC(I)=NCC(I) GOTO 4650
```

```
4630 PRINT #1, "/";
4640 PRINT #1, USING F3$;NCC(I);
4650 PRINT #1, TAB(63);
4660 IF B$(I)="O" THEN PRINT #1, "-"
4670 IF B$(I)="O" GOTO 5190
4680 IF BSS(I)<>0 GOTO 4730
4690 IF BSF(I)<>0 GOTO 4770
4700 IF BFF(I)<>0 GOTO 4810
4710 PRINT #1, B$(I);
4720 GOTO 4840
4730 PRINT #1, B$(I);"(SS=";
4740 PRINT #1, USING F2$;BSS(I);
4750 PRINT #1, ")";
4760 GOTO 4840
4770 PRINT #1, B$(I);"(SF=";
4780 PRINT #1, USING F2$;BSF(I);
4790 PRINT #1, ")";
4800 GOTO 4840
4810 PRINT #1, B$(I);"(FF=";
4820 PRINT #1, USING F2$;BFF(I);
4830 PRINT #1, ")";
4840 IF C$(I)="O" GOTO 5180
4850 IF CSS(I)<>0 GOTO 4900
4860 IF CSF(I)<>0 GOTO 4940
4870 IF CFF(I)<>0 GOTO 4980
4880 PRINT #1, ",";C$(I);
4890 GOTO 5010
4900 PRINT #1, ",";C$(I);"(SS=";
4910 PRINT #1, USING F2$;CSS(I);
4920 PRINT #1, ")";
4930 GOTO 5010
4940 PRINT #1, ",";C$(I);"(SF=";
4950 PRINT #1, USING F2$;CSF(I);
4960 PRINT #1, ")";
4970 GOTO 5010
4980 PRINT #1, ",";C$(I);"(FF=";
4990 PRINT #1, USING F2$;CFF(I);
5000 PRINT #1, ")";
5010 IF D$(I)="O" GOTO 5180
5020 IF DSS(I)<>0 GOTO 5070
5030 IF DSF(I)<>0 GOTO 5110
5040 IF DFF(I)<>0 GOTO 5150
5050 PRINT #1, ",";D$(I)
5060 GOTO 5190
5070 PRINT #1, ",";D$(I);"(SS=";
5080 PRINT #1, USING F2$;DSS(I);
5090 PRINT #1, ")";
5100 GOTO 5180
5110 PRINT #1, ",";D$(I);"(SF=";
5120 PRINT #1, USING F2$;DSF(I);
5130 PRINT #1, ")";
5140 GOTO 5180
```

```
5150 PRINT #1, ",";D$(I);"(FF=";
5160 PRINT #1, USING F2$;DFF(I);
5170 PRINT #1, ")";
5180 PRINT #1, " "
5190 NEXT I
5200 IF INDCOST=0 GOTO 5270
5210 PRINT #1, " "
5220 PRINT #1, " "
5230 IF J$(4)="1" THEN T$="HOUR"
5240 IF J$(4)="2" THEN T$="DAY"
5250 IF J$(4)="3" THEN T$="WEEK"
5260 PRINT #1, TAB(6);"PROJECT INDIRECT COST= $";INDCOST;"PER ";T$
5270 IF YY$<>"Y" GOTO 5290
5280 PRINT #1, CHR$(12)
5290 CLOSE #1
5300 IF YY$<>"Y" GOTO 5380
5310 PRINT " "
5320 INPUT "DO YOU WANT ANOTHER HARD COPY OF THIS DATA (Y OR N)";GG$
5330 CLS
5340 IF GG$="Y" THEN 4310 ELSE 5470
5350 REM
5360 REM      DO YOU WANT TO EDIT/HARDCOPY OF INPUT DATA
5370 REM
5380 PRINT " "
5390 INPUT "DO YOU WANT TO EDIT? Y OR N";E$
5400 IF E$="Y" GOTO 7500
5410 INPUT "DO YOU WANT A HARD COPY OF THE INPUT DATA (Y OR N)";YY$
5420 CLS
5430 IF YY$="Y" GOTO 4310
5440 REM
5450 REM      PRINTING CPM SCHEDULE TABLE
5460 REM
5470 EE$="N"
5480 DEVICE$="SCRN:":WIDTH "SCRN:",80
5490 GOTO 5510
5500 DEVICE$="LPT1:":WIDTH "LPT1:",80
5510 OPEN DEVICE$ FOR OUTPUT AS #1
5520 IF OPTCOST=0 GOTO 5550
5530 PRINT #1, TAB(27);"OPTIMUM COST COMPLETION TIME"
5540 GOTO 5590
5550 IF GH<>1 GOTO 5580
5560 PRINT #1, TAB(30);"CRASHED COMPLETION TIME"
5570 GOTO 5590
5580 PRINT #1, TAB(30);"NORMAL COMPLETION TIME"
5590 PRINT #1, TAB(34);"CPM SCHEDULING"
5600 PRINT #1, " "
5610 PRINT #1, TAB(34);"PROJECT: ";J$(1)
5620 IF J$(4)="1" THEN T$="HOURS"
5630 IF J$(4)="2" THEN T$="DAYS"
5640 IF J$(4)="3" THEN T$="WEEKS"
5650 PRINT #1, TAB(34);"TIME SCALE: ";T$
5660 PRINT #1, ""
```

```
5670 PRINT #1, ""
5680 PRINT #1, TAB(4);"ACT.";TAB(11);"ACTIVITY";TAB(27);"ACT.";TAB(36);"EARLY";T
AB(45);"EARLY";TAB(54);"LATEST";TAB(63);"LATEST";TAB(72);"TOTAL"
5690 PRINT #1, TAB(4);"CODE";TAB(11);"DESCRIPTION";TAB(27);"DUR.";TAB(36);"START
";TAB(45);"FINISH";TAB(54);"START";TAB(63);"FINISH";TAB(72);"FLOAT";TAB(78);"CP"

5700 PRINT #1, TAB(4);"----";TAB(11);"-----------";TAB(27);"----";TAB(36);"-----
";TAB(45);"------";TAB(54);"-----";TAB(63);"------";TAB(72);"-----";TAB(78);"--"

5710 LL=0
5720 FOR I=1 TO N
5730 IF EE$="Y" GOTO 5780
5740 LL=LL+1
5750 IF LL<>15 GOTO 5780
5760 LL=0
5770 INPUT "HIT THE ENTER KEY TO CONTINUE";WW$
5780 F1$="###.#"
5790 PRINT #1, TAB(5);A$(I);TAB(12);AD$(I);TAB(26);
5800 PRINT #1, USING F1$;D(I);
5810 PRINT #1, TAB(35);
5820 PRINT #1, USING F1$;ES(I);
5830 PRINT #1, TAB(44);
5840 PRINT #1, USING F1$;EF(I);
5850 PRINT #1, TAB(53);
5860 PRINT #1, USING F1$;LS(I);
5870 PRINT #1, TAB(62);
5880 PRINT #1, USING F1$;LF(I);
5890 PRINT #1, TAB(71);
5900 PRINT #1, USING F1$;F(I);
5910 PRINT #1, TAB(78);
5920 PRINT #1, CP$(I)
5930 NEXT I
5940 PRINT #1, " "
5950 IF EE$="Y" GOTO 5980
5960 IF LL<10 GOTO 5980
5970 INPUT "HIT THE ENTER KEY TO CONTINUE";WW$
5980 PRINT #1, TAB(4);"PROJECT DURATION: ";TT
5990 IF DIRC=0 GOTO 6080
6000 PRINT #1, " "
6010 PRINT #1, TAB(4);"PROJECT DIRECT COST: $";DIRC
6020 PRINT #1, " "
6030 IF ICD$<>"Y" GOTO 6080
6040 PRINT #1, TAB(4);"PROJECT INDIRECT COST: $";INDCT
6050 PRINT #1, " "
6060 TCOST=DIRC+INDCT
6070 PRINT #1, TAB(4);"PROJECT TOTAL COST: $";TCOST
6080 IF EE$="Y" GOTO 6110
6090 PRINT " "
6100 INPUT "HIT THE ENTER KEY WHEN READY TO PROCEED TO THE BAR CHART";UU$
6110 PRINT #1, CHR$(12)
6120 REM
6130 REM      PRINT BAR CHART
6140 REM
6150 IF OPTCOST=0 GOTO 6180
6160 PRINT #1, TAB(28);"OPTIMUM COST COMPLETION TIME"
```

```
6170 GOTO 6220
6180 IF GH<>1 GOTO 6210
6190 PRINT #1, TAB(31);"CRASHED COMPLETION TIME"
6200 GOTO 6220
6210 PRINT #1, TAB(31);"NORMAL COMPLETION TIME"
6220 PRINT #1, TAB(34);"BAR (GANNT) CHART"
6230 PRINT #1, " "
6240 PRINT #1, TAB(34);"PROJECT: ";J$(1)
6250 PRINT #1, TAB(34);"TIME SCALE: ";T$
6260 PRINT #1, " "
6270 PRINT #1, " "
6280 IF TT<27 THEN F1=2
6290 IF (TT>=27) AND (TT<55) THEN F1=1
6300 IF (TT>=55) AND (TT<110) THEN F1=.5
6310 IF (TT>=110) AND (TT<220) THEN F1=.25
6320 IF (TT>=220) AND (TT<440) THEN F1=.125
6330 IF F1=2 GOTO 7370
6340 IF F1=1 GOTO 6380
6350 IF F1=.5 GOTO 7330
6360 IF F1=.25 GOTO 7400
6370 IF F1=.125 GOTO 7440
6380 PRINT #1, TAB(4);"ACT.";TAB(11);"ACTIVITY";TAB(35);"1";TAB(45);"2";TAB(55);
"3";TAB(65);"4";TAB(75);"5"
6390 PRINT #1, TAB(4);"CODE";TAB(11);"DESCRIPTION";TAB(25);"0";TAB(35);"0";TAB(4
5);"0";TAB(55);"0";TAB(65);"0";TAB(75);"0"
6400 PRINT #1, TAB(4);STRING$(21,CHR$(205));CHR$(216);STRING$(9,CHR$(205));CHR$(
216);
6410 PRINT #1, STRING$(9,CHR$(205));CHR$(216);STRING$(9,CHR$(205));CHR$(216);
6420 PRINT #1, STRING$(9,CHR$(205));CHR$(216);STRING$(9,CHR$(205));CHR$(216);
6430 PRINT #1, STRING$(4,CHR$(205))
6440 LL=0
6450 FOR I=1 TO N
6460 IF EE$="Y" GOTO 6520
6470 LL=LL+1
6480 IF LL<>15 GOTO 6530
6490 LL=0
6500 INPUT "HIT THE ENTER KEY TO CONTINUE";WW$
6510 GOTO 6530
6520 PRINT #1, TAB(25);CHR$(186)
6530 DD1=D(I)*F1
6540 IF DD1<1 THEN DD1=1
6550 EE1=F(I)*F1
6560 IF EE1<1 THEN EE1=1
6570 IF F(I)=0 GOTO 6750
6580 D1=D(I)*F1
6590 D2=INT(D(I)*F1)
6600 D3=F(I)*F1
6610 D4=INT(F(I)*F1)
6620 IF (D1<>D2) AND (D3<>D4) GOTO 6690
6630 PRINT #1, TAB(5);A$(I);TAB(11);AD$(I);TAB(25);CHR$(186);
6640 COLOR 1
6650 PRINT #1, TAB(INT(26+ES(I)*F1));STRING$(INT(DD1),CHR$(178));
6660 COLOR 14
6670 PRINT #1, STRING$(INT(EE1),CHR$(176))
```

```
6680 GOTO 6780
6690 PRINT #1, TAB(5);A$(I);TAB(11);AD$(I);TAB(25);CHR$(186);
6700 COLOR 1
6710 PRINT #1, TAB(INT(26+ES(I)*F1));STRING$(INT(DD1),CHR$(178));
6720 COLOR 14
6730 PRINT #1, STRING$(INT(EE1+1),CHR$(176))
6740 GOTO 6780
6750 PRINT #1, TAB(5);A$(I);TAB(11);AD$(I);TAB(25);CHR$(186);
6760 COLOR 4
6770 PRINT #1, TAB(INT(26+ES(I)*F1));STRING$(INT(DD1),CHR$(219))
6780 COLOR 7
6790 NEXT I
6800 PRINT #1, " "
6810 COLOR 4
6820 PRINT #1, TAB(4);CHR$(219);
6830 COLOR 7
6840 PRINT #1, "DENOTES CRITICAL ACTIVITY"
6850 COLOR 1
6860 PRINT #1, TAB(4);CHR$(178);
6870 COLOR 7
6880 PRINT #1, "DENOTES NON-CRITICAL ACTIVITY"
6890 COLOR 14
6900 PRINT #1, TAB(4);CHR$(176);
6910 COLOR 7
6920 PRINT #1, "DENOTES FLOAT TIME"
6930 IF EE$<>"Y" GOTO 6950
6940 PRINT #1, CHR$(12)
6950 PRINT " "
6960 CLOSE #1
6970 IF EE$="Y" GOTO 7010
6980 INPUT "DO YOU WISH TO GET A HARD COPY OF PREVIOUS CHARTS (Y OR N)";EE$
6990 CLS
7000 IF EE$="Y" THEN 5500 ELSE 7030
7010 INPUT "DO YOU WISH TO GET ANOTHER HARD COPY OF THE CHARTS (Y OR N)";GG$
7020 IF GG$="Y" GOTO 5500
7030 IF GH=1 GOTO 14200
7040 IF OPTCOST=1 GOTO 14340
7050 PRINT " "
7060 INPUT "DO YOU WANT THE PROJECT DATA STORED INTO A FILE (Y OR N)";BB$
7070 IF BB$<>"Y" GOTO 7090
7080 GOSUB 9640
7090 PRINT " "
7100 INPUT "DO YOU WISH TO PERFORM A COMPRESSION ANALYSIS ON THE PROJECT (Y OR N
)";VV$
7110 IF VV$<>"Y" GOTO 7290
7120 PRINT " "
7130 INPUT "DO YOU NEED TO ADD COMPRESSION DATA TO ACTIVITY DATA ALREADY FILED (
Y OR N)";AZ$
7140 IF AZ$="Y" GOTO 7260
7150 CLS
7160 PRINT "DUE TO THE ITERATIVE NATURE OF THE COMPRESSION CALCULATIONS"
7170 PRINT "THE COMPUTER MAY BE SLOW IN PRODUCING THE FOLLOWING TABLE."
7180 PRINT "TO REDUCE TIME:  IF THE INPUTED NETWORK IS A COMPLICATED ONE"
```

```
7190 PRINT "AND A HARD COPY IS DESIRED IT IS BEST TO ELIMINATE THE SCREEN"
7200 PRINT "EDITION AND ONLY PRODUCE THE HARD COPY"
7210 PRINT " "
7220 INPUT "DO YOU WISH THE OUTPUT TO GO DIRECTLY TO THE PRINTER (Y OR N)";WAW$
7230 PRINT " "
7240 PRINT "REMEMBER IT MAY TAKE A LONG TIME DEPENDING ON THE NETWORK"
7250 INPUT "HIT THE ENTER KEY TO PROCEED";CVB$
7260 CLS
7270 IF AZ$="Y" GOTO 7850
7280 GOTO 9810
7290 PRINT " "
7300 INPUT "DO YOU WANT TO END THE PROGRAM";EN$
7310 IF EN$<>"Y" GOTO 5390
7320 END
7330 PRINT #1, TAB(75);"1"
7340 PRINT #1, TAB(4);"ACT.";TAB(11);"ACTIVITY";TAB(35);"2";TAB(45);"4";TAB(55);
"6";TAB(65);"8";TAB(75);"0"
7350 PRINT #1, TAB(4);"CODE";TAB(11);"DESCRIPTION";TAB(25);"0";TAB(35);"0";TAB(4
5);"0";TAB(55);"0";TAB(65);"0";TAB(75);"0"
7360 GOTO 6400
7370 PRINT #1, TAB(4);"ACT.";TAB(11);"ACTIVITY";TAB(45);"1";TAB(55);"1";TAB(65);
"2";TAB(75);"2"
7380 PRINT #1, TAB(4);"CODE";TAB(11);"DESCRIPTION";TAB(25);"0";TAB(35);"5";TAB(4
5);"0";TAB(55);"5";TAB(65);"0";TAB(75);"5"
7390 GOTO 6400
7400 PRINT #1, TAB(55);"1";TAB(65);"1";TAB(75);"2"
7410 PRINT #1, TAB(4);"ACT.";TAB(11);"ACTIVITY";TAB(35);"4";TAB(45);"8";TAB(55);
"2";TAB(65);"6";TAB(75);"0"
7420 PRINT #1, TAB(4);"CODE";TAB(11);"DESCRIPTION";TAB(25);"0";TAB(35);"0";TAB(4
5);"0";TAB(55);"0";TAB(65);"0";TAB(75);"0"
7430 GOTO 6400
7440 PRINT #1, TAB(45);"1";TAB(55);"2";TAB(65);"3";TAB(75);"4"
7450 PRINT #1, TAB(4);"ACT.";TAB(11);"ACTIVITY";TAB(35);"8";TAB(45);"6";TAB(55);
"4";TAB(65);"2";TAB(75);"0"
7460 PRINT #1, TAB(4);"CODE";TAB(11);"DESCRIPTION";TAB(25);"0";TAB(35);"0";TAB(4
5);"0";TAB(55);"0";TAB(65);"0";TAB(75);"0"
7470 GOTO 6400
7480 REM     EDITING PHASE
7490 REM
7500 INPUT "DO YOU WISH TO CHANGE PROJECT HEADLINE DATA (Y OR N)";BNM$
7510 IF BNM$<>"Y" GOTO 7580
7520 INPUT "ENTER THE NAME OF WHO IS SCHEDULING THE PROJECT";J$(2)
7530 INPUT "ENTER TODAY'S DATE";J$(3)
7540 INPUT "ENTER THE PROJECT NAME";J$(1)
7550 INPUT "ENTER THE UNIT OF TIME. 1=HOURS  2=DAYS  3=WEEKS";J$(4)
7560 CLS
7570 GOTO 4280
7580 INPUT "DO YOU WISH TO INPUT ADVANCE PRECEDENCE CAPABILITIES WHILE EDITING";
KK$
7590 INPUT "DO YOU WISH TO INPUT OR INSURE RETAINMENT OF ACTIVITY DIRECT COST DA
TA";ACTDC$
```

```
7600 INPUT "DO YOU WISH TO INPUT OR INSURE THE RETAINMENT OF COMPRESSION DATA";P
P$
7610 INPUT "DO YOU WISH TO INCLUDE OR CHANGE THE INDIRECT PROJECT COST (Y OR N)"
;UNIT$
7620 IF UNIT$<>"Y" GOTO 7690
7630 INPUT "ENTER THE INDIRECT COST PER UNIT OF TIME";INDCOST
7640 ICD$="Y"
7650 IF INDCOST=0 THEN ICD$="N"
7660 INDCT=INDCOST*TT
7670 CLS
7680 GOTO 4280
7690 INPUT "DO YOU WISH TO ADD ACTIVITIES? Y OR N";ZQ$
7700 IF ZQ$="Y" THEN CLS
7710 IF ZQ$="Y" GOTO 8570
7720 INPUT "DO YOU WISH TO DELETE ACTIVITIES? Y OR N";QZ$
7730 IF QZ$="Y" THEN CLS
7740 IF QZ$="Y" GOTO 9130
7750 INPUT "DO YOU WISH TO CHANGE AN ACTIVITY? Y OR N";QQ$
7760 CLS
7770 IF QQ$="Y" GOTO 7840
7780 IF ZQ$="Y" GOTO 1170
7790 IF QZ$="Y" GOTO 1170
7800 GOTO 7060
7810 REM
7820 REM     CHANGING ACTIVITY DATA
7830 REM
7840 AZ$="N"
7850 PRINT"ENTER ACTIVITY CODE OF ACTIVITY TO BE CHANGED"
7860 INPUT Q$
7870 FOR I=1 TO N
7880 IF A$(I)=Q$ GOTO 7920
7890 NEXT I
7900 GOTO 1170
7910 PRINT "RE-ENTER THE DATA"
7920 INPUT "ENTER NEW ACTIVITY CODE";A$(I)
7930 INPUT "ENTER NEW ACTIVITY DESCRIPTION";AD$(I)
7940 IF PP$<>"Y" GOTO 8000
7950 INPUT "ENTER THE NEW NORMAL TIME DURATION";D(I)
7960 INPUT "ENTER THE NEW CRASH TIME DURATION";CD(I)
7970 INPUT "ENTER THE NEW NORMAL COST";NNC(I)
7980 INPUT "ENTER THE NEW CRASH COST";NCC(I)
7990 GOTO 8050
8000 INPUT "ENTER NEW ACTIVITY DURATION";D(I)
8010 CD(I)=D(I)
8020 IF ACTDC$<>"Y" GOTO 8050
8030 INPUT "ENTER NEW ACTIVITY COST";NNC(I)
8040 NCC(I)=NNC(I)
8050 B$(I)="0"
8060 C$(I)="0"
8070 D$(I)="0"
8080 BSS(I)=0
8090 CSS(I)=0
8100 DSS(I)=0
```

```
8110 BSF(I)=0
8120 CSF(I)=0
8130 DSF(I)=0
8140 BFF(I)=0
8150 CFF(I)=0
8160 DFF(I)=0
8170 INPUT "ENTER NEW DEPENDENCY 1. IF NONE, ENTER: O";B$(I)
8180 IF B$(I)="O" GOTO 8470
8190 IF KK$<>"Y" GOTO 8270
8200 INPUT "IF THE DEPENDENCY IS NORMAL ENTER: O.  ELSE ENTER: SF, SS, OR FF";AA
$
8210 IF AA$<>"SF" GOTO 8230
8220 INPUT "ENTER THE START-TO-FINISH TIME";BSF(I)
8230 IF AA$<>"SS" GOTO 8250
8240 INPUT "ENTER THE START-TO-START TIME";BSS(I)
8250 IF AA$<>"FF" GOTO 8270
8260 INPUT "ENTER THE FINISH-TO-FINISH TIME";BFF(I)
8270 INPUT "ENTER NEW DEPENDENCY 2. IF NONE, ENTER: O";C$(I)
8280 IF C$(I)="O" GOTO 8470
8290 IF KK$<>"Y" GOTO 8370
8300 INPUT "IF THE DEPENDENCY IS NORMAL ENTER: O.  ELSE ENTER: SF, SS, OR FF";AA
$
8310 IF AA$<>"SF" GOTO 8330
8320 INPUT "ENTER THE START-TO-FINISH TIME";CSF(I)
8330 IF AA$<>"SS" GOTO 8350
8340 INPUT "ENTER THE START-TO-START TIME";CSS(I)
8350 IF AA$<>"FF" GOTO 8370
8360 INPUT "ENTER THE FINISH-TO-FINISH TIME";CFF(I)
8370 INPUT "ENTER NEW DEPENDENCY 3. IF NONE, ENTER: O";D$(I)
8380 IF D$(I)="O" GOTO 8470
8390 IF KK$<>"Y" GOTO 8470
8400 INPUT "IF THE DEPENDENCY IS NORMAL ENTER: O.  ELSE ENTER: SF, SS, OR FF";AA
$
8410 IF AA$<>"SF" GOTO 8430
8420 INPUT "ENTER THE START-TO-FINISH TIME";DSF(I)
8430 IF AA$<>"SS" GOTO 8450
8440 INPUT "ENTER THE START-TO-START TIME";DSS(I)
8450 IF AA$<>"FF" GOTO 8470
8460 INPUT "ENTER THE FINISH-TO-FINISH TIME";DFF(I)
8470 INPUT "IS THE DATA CORRECT? Y OR N";Z$
8480 IF Z$="N" GOTO 7910
8490 INPUT "DO YOU WISH TO CHANGE ANOTHER ACTIVITY? Y OR N";TQ$
8500 CLS
8510 IF TQ$="Y" GOTO 7850
8520 IF AZ$="Y" GOTO 7060
8530 GOTO 1170
8540 REM
8550 REM      ADDING ACTIVITIES
8560 REM
8570 INPUT "ENTER NUMBER OF ACTIVITIES TO BE ADDED";G
8580 FOR I=1 TO G
8590 INPUT "ENTER ACTIVITY CODE FOR ACTIVITY";A$(N+I)
8600 INPUT "ENTER ACTIVITY DESCRIPTION";AD$(N+I)
8610 IF PP$<>"Y" GOTO 8670
```

```
8620 INPUT "ENTER THE NORMAL TIME DURATION";D(N+I)
8630 INPUT "ENTER THE CRASH TIME DURATION";CD(N+I)
8640 INPUT "ENTER THE NORMAL COST";NNC(N+I)
8650 INPUT "ENTER THE CRASH COST";NCC(N+I)
8660 GOTO 8720
8670 INPUT "ENTER ACTIVITY DURATION";D(N+I)
8680 CD(N+I)=D(N+I)
8690 IF ACTDC$<>"Y" GOTO 8720
8700 INPUT "ENTER ACTIVITY COST";NNC(N+I)
8710 NCC(N+I)=NNC(N+I)
8720 INPUT "ENTER DEPENDENCY 1.  IF NONE, ENTER: 0";B$(N+I)
8730 IF B$(N+I)="0" GOTO 9020
8740 IF KK$<>"Y" GOTO 8820
8750 INPUT "IF THE DEPENDENCY IS NORMAL ENTER: 0.  ELSE ENTER: SF, SS, OR FF";AA
$
8760 IF AA$<>"SF" GOTO 8780
8770 INPUT "ENTER THE START-TO-FINISH TIME";BSF(N+I)
8780 IF AA$<>"SS" GOTO 8800
8790 INPUT "ENTER THE START-TO-START TIME";BSS(N+I)
8800 IF AA$<>"FF" GOTO 8820
8810 INPUT "ENTER THE FINISH-TO-FINISH TIME";BFF(N+I)
8820 INPUT "ENTER DEPENDENCY 2.  IF NONE, ENTER: 0";C$(N+I)
8830 IF C$(N+I)="0" GOTO 9020
8840 IF KK$<>"Y" GOTO 8920
8850 INPUT "IF THE DEPENDENCY IS NORMAL ENTER: 0.  ELSE ENTER: SF, SS, OR FF";AA
$
8860 IF AA$<>"SF" GOTO 8880
8870 INPUT "ENTER THE START-TO-FINISH TIME";CSF(N+I)
8880 IF AA$<>"SS" GOTO 8900
8890 INPUT "ENTER THE START-TO-START TIME";CSS(N+I)
8900 IF AA$<>"FF" GOTO 8920
8910 INPUT "ENTER THE FINISH-TO-FINISH TIME";CFF(N+I)
8920 INPUT "ENTER DEPENDENCY 3.  IF NONE, ENTER: 0";D$(N+I)
8930 IF D$(N+I)="0" GOTO 9020
8940 IF KK$<>"Y" GOTO 9020
8950 INPUT "IF THE DEPENDENCY IS NORMAL ENTER: 0.  ELSE ENTER: SF, SS, OR FF";AA
$
8960 IF AA$<>"SF" GOTO 8980
8970 INPUT "ENTER THE START-TO-FINISH TIME";DSF(N+I)
8980 IF AA$<>"SS" GOTO 9000
8990 INPUT "ENTER THE START-TO-START TIME";DSS(N+I)
9000 IF AA$<>"FF" GOTO 9020
9010 INPUT "ENTER THE FINISH-TO-FINISH TIME";DFF(N+I)
9020 INPUT "IS THE DATA CORRECT? Y OR N";P$
9030 IF P$="Y" GOTO 9060
9040 PRINT "RE-ENTER THE DATA"
9050 GOTO 8590
9060 CLS
9070 NEXT I
9080 N=N+G
9090 GOTO 7720
9100 REM
9110 REM     DELETING ACTIVITIES
9120 REM
```

```
9130 INPUT "ENTER ACTIVITY CODE TO BE DELETED";JINX$
9140 FOR I=1 TO N
9150 IF A$(I)=JINX$ THEN 9160 ELSE NEXT I
9160 IF I=N GOTO 9370
9170 FOR II=I TO N-1
9180 A$(II)=A$(II+1)
9190 AD$(II)=AD$(II+1)
9200 D(II)=D(II+1)
9210 CD(II)=CD(II+1)
9220 NCC(II)=NCC(II+1)
9230 NNC(II)=NNC(II+1)
9240 B$(II)=B$(II+1)
9250 C$(II)=C$(II+1)
9260 D$(II)=D$(II+1)
9270 BSS(II)=BSS(II+1)
9280 CSS(II)=CSS(II+1)
9290 DSS(II)=DSS(II+1)
9300 BSF(II)=BSF(II+1)
9310 CSF(II)=CSF(II+1)
9320 DSF(II)=DSF(II+1)
9330 BFF(II)=BFF(II+1)
9340 CFF(II)=CFF(II+1)
9350 DFF(II)=DFF(II+1)
9360 NEXT II
9370 N=N-1
9380 INPUT "DELETE ANOTHER? Y OR N";X$
9390 CLS
9400 IF X$="N" GOTO 7750
9410 GOTO 9130
9420 REM
9430 REM     SUBROUTINE TO READ BACK DATA FROM A FILE
9440 REM
9450 CLS
9460 OPEN "I",#1,F$
9470 FOR I=1 TO 4
9480 INPUT #1, J$(I)
9490 NEXT I
9500 INPUT #1, N
9510 INPUT #1, TT
9520 INPUT #1, DIRC
9530 INPUT #1, INDCOST
9540 FOR I=1 TO N
9550 INPUT #1, A$(I),AD$(I),B$(I),C$(I),D$(I),D(I),ES(I),EF(I),LS(I),LF(I)
9560 INPUT #1, CP$(I),F(I),BSS(I),CSS(I),DSS(I),BSF(I),CSF(I),DSF(I)
9570 INPUT #1, BFF(I),CFF(I),DFF(I),CD(I),NNC(I),NCC(I)
9580 NEXT I
9590 CLOSE #1
9600 RETURN
9610 REM
9620 REM     SUBROUTINE TO READ DATA INTO A FILE
9630 REM
9640 CLS
9650 F$=J$(1)
```

```
9660 OPEN "O",#1,F$
9670 FOR I=1 TO 4
9680 WRITE #1, J$(I)
9690 NEXT I
9700 WRITE #1, N
9710 WRITE #1, TT
9720 WRITE #1, DIRC
9730 WRITE #1, INDCOST
9740 FOR I=1 TO N
9750 WRITE #1, A$(I),AD$(I),B$(I),C$(I),D$(I),D(I),ES(I),EF(I),LS(I),LF(I)
9760 WRITE #1, CP$(I),F(I),BSS(I),CSS(I),DSS(I),BSF(I),CSF(I),DSF(I)
9770 WRITE #1, BFF(I),CFF(I),DFF(I),CD(I),NNC(I),NCC(I)
9780 NEXT I
9790 CLOSE #1
9800 RETURN
9810 REM
9820 REM     COMPRESSION ANALYSIS CALCULATIONS
9830 REM
9840 YB=YB+1
9850 IF YB>1 GOTO 9940
9860 CHANGE=DIRC
9870 DRC=DIRC
9880 FOR I=1 TO N
9890 XD(I)=D(I)
9900 NCT(I)=NNC(I)
9910 NEXT I
9920 T1=TT
9930 NINDCT=INDCT
9940 XB=XB+1
9950 IF XB>1 GOTO 10110
9960 TM=TT-1
9970 FOR I=1 TO N
9980 TCS(I)=0
9990 IJ(I)=0
10000 JI(I)=0
10010 NEXT I
10020 K=0
10030 FOR I=1 TO N
10040 IF CP$(I)<>"C" GOTO 10090
10050 IF XD(I)=CD(I) GOTO 10090
10060 TCS(I)=((NCC(I)-NCT(I))/(XD(I)-CD(I)))
10070 TCS(I)=INT(TCS(I))
10080 K=K+1
10090 NEXT I
10100 REM     SORT BY CRASH CAPABILITY
10110 FOR NM=1 TO N
10120 FOR I=NM+1 TO N
10130 IF TCS(I)=0 GOTO 11000
10140 IF TCS(NM)=0 GOTO 10300
10150 IF TCS(NM)>TCS(I) GOTO 10300
10160 IF TCS(NM)<>TCS(I) GOTO 11000
10170 ER=ER+1
10180 IF EE(NM)<>0 OR EE(I)<>0 GOTO 10230
```

```
10190 EE(NM)=ER
10200 ER=ER+1
10210 EE(I)=ER
10220 GOTO 11000
10230 IF EE(I)<>0 GOTO 10260
10240 EE(I)=ER
10250 GOTO 11000
10260 IF EE(NM)<>0 GOTO 10290
10270 EE(NM)=ER
10280 GOTO 10300
10290 IF EE(NM)<EE(I) THEN 11000 ELSE 10300
10300 AA$=A$(I)
10310 ADAD$=AD$(I)
10320 DD=D(I)
10330 BB$=B$(I)
10340 CC$=C$(I)
10350 DD$=D$(I)
10360 CD1=CD(I)
10370 NNC1=NNC(I)
10380 NCC1=NCC(I)
10390 XD1=XD(I)
10400 TCS1=TCS(I)
10410 IJ1=IJ(I)
10420 JI1=JI(I)
10430 CCD1=CCD(I)
10440 BSSS=BSS(I)
10450 CSSS=CSS(I)
10460 DSSS=DSS(I)
10470 BSFF=BSF(I)
10480 CSFF=CSF(I)
10490 DSFF=DSF(I)
10500 BFFF=BFF(I)
10510 CFFF=CFF(I)
10520 DFFF=DFF(I)
10530 EE1=EE(I)
10540 EE(I)=EE(NM):EE(NM)=EE1
10550 A$(I)=A$(NM):A$(NM)=AA$
10560 AD$(I)=AD$(NM):AD$(NM)=ADAD$
10570 D(I)=D(NM):D(NM)=DD
10580 B$(I)=B$(NM):B$(NM)=BB$
10590 C$(I)=C$(NM):C$(NM)=CC$
10600 D$(I)=D$(NM):D$(NM)=DD$
10610 CD(I)=CD(NM):CD(NM)=CD1
10620 NNC(I)=NNC(NM):NNC(NM)=NNC1
10630 NCC(I)=NCC(NM):NCC(NM)=NCC1
10640 XD(I)=XD(NM):XD(NM)=XD1
10650 TCS(I)=TCS(NM):TCS(NM)=TCS1
10660 IJ(I)=IJ(NM):IJ(NM)=IJ1
10670 JI(I)=JI(NM):JI(NM)=JI1
10680 CCD(I)=CCD(NM):CCD(NM)=CCD1
10690 BSS(I)=BSS(NM):BSS(NM)=BSSS
10700 CSS(I)=CSS(NM):CSS(NM)=CSSS
10710 DSS(I)=DSS(NM):DSS(NM)=DSSS
```

```
10720 BSF(I)=BSF(NM):BSF(NM)=BSFF
10730 CSF(I)=CSF(NM):CSF(NM)=CSFF
10740 DSF(I)=DSF(NM):DSF(NM)=DSFF
10750 BFF(I)=BFF(NM):BFF(NM)=BFFF
10760 CFF(I)=CFF(NM):CFF(NM)=CFFF
10770 DFF(I)=DFF(NM):DFF(NM)=DFFF
10780 ESES=ES(I)
10790 EFEF=EF(I)
10800 LSLS=LS(I)
10810 LFLF=LF(I)
10820 CPCP$=CP$(I)
10830 FF=F(I)
10840 ES(I)=ES(NM):ES(NM)=ESES
10850 EF(I)=EF(NM):EF(NM)=EFEF
10860 LS(I)=LS(NM):LS(NM)=LSLS
10870 LF(I)=LF(NM):LF(NM)=LFLF
10880 CP$(I)=CP$(NM):CP$(NM)=CPCP$
10890 F(I)=F(NM):F(NM)=FF
10900 NCT1=NCT(I)
10910 NPT1=NPT(I)
10920 DPT1=DPT(I)
10930 HH1=HH(I)
10940 JKL1=JKL(I)
10950 NCT(I)=NCT(NM):NCT(NM)=NCT1
10960 NPT(I)=NPT(NM):NPT(NM)=NPT1
10970 DPT(I)=DPT(NM):DPT(NM)=DPT1
10980 HH(I)=HH(NM):HH(NM)=HH1
10990 JKL(I)=JKL(NM):JKL(NM)=JKL1
11000 NEXT I
11010 NEXT NM
11020 REM
11030 REM      COMPRESSION ANALYSIS CALCULATIONS CONTINUED
11040 REM
11050 IF XL=1 GOTO 12710
11060 IF ZL=1 GOTO 12640
11070 IF SV>0 GOTO 12540
11080 IF PD=1 GOTO 12480
11090 IF RM=1 GOTO 12040
11100 IF GB=1 GOTO 11970
11110 IF VS>0 GOTO 11870
11120 IF ED=1 GOTO 11810
11130 IF XK=1 GOTO 11430
11140 IF YK=1 GOTO 11360
11150 IF XB=1 GOTO 11180
11160 IF TT=TM THEN 11250
11170 IF XB>1 GOTO 11230
11180 FOR IT=1 TO K
11190 IF D(IT)=CD(IT) THEN CCD(IT)=1
11200 IF D(IT)>CD(IT) THEN D(IT)=D(IT)-1 ELSE 11230
11210 NNC(IT)=NNC(IT)+TCS(IT)
11220 GOTO 1170
11230 NEXT IT
```

```
11240 GOTO 13160
11250 IJ(IT)=IT
11260 PI=IT
11270 HML=1
11280 IF PI=1 GOTO 11520
11290 FOR J=1 TO PI-1
11300 IF CCD(J)=1 GOTO 11330
11310 D(J)=D(J)+1
11320 NNC(J)=NNC(J)-TCS(J)
11330 NEXT J
11340 YK=1
11350 GOTO 1170
11360 IF TT=TM GOTO 11520
11370 FOR JQ=1 TO PI-1
11380 IF CCD(JQ)=1 GOTO 11510
11390 D(JQ)=D(JQ)-1
11400 NNC(JQ)=NNC(JQ)+TCS(JQ)
11410 XK=1
11420 GOTO 1170
11430 IF TT=TM THEN PI=JQ ELSE 11490
11440 XK=0
11450 HML=0
11460 IJ(JQ)=JQ
11470 IF PI=1 GOTO 11520
11480 GOTO 11290
11490 IF HML=0 GOTO 11510
11500 QJJ=JQ+1
11510 NEXT JQ
11520 STOTAL=0
11530 FOR QI=1 TO K
11540 STOTAL=STOTAL+TCS(IJ(QI))
11550 NEXT QI
11560 YV=0
11570 CZ=0
11580 IF IT=<3 GOTO 11650
11590 IF QJJ=0 GOTO 11650
11600 FOR TCM=QJJ+1 TO IT-1
11610 IF TCS(TCM)+TCS(IT)=>STOTAL GOTO 11640
11620 YV=YV+1
11630 NEXT TCM
11640 CZ=QJJ+YV
11650 FOR JL=1 TO K
11660 IF IJ(JL)=0 GOTO 11690
11670 D(JL)=D(JL)+1
11680 NNC(JL)=NNC(JL)-TCS(JL)
11690 NEXT JL
11700 IF YV=0 GOTO 12340
11710 ED=1
11720 JKL(IT)=IT
11730 D(IT)=D(IT)-1
11740 NNC(IT)=NNC(IT)+TCS(IT)
11750 FOR SCN=QJJ+1 TO CZ
```

```
11760 IF D(SCN)=CD(SCN) THEN CCD(SCN)=1
11770 IF D(SCN)>CD(SCN) THEN D(SCN)=D(SCN)-1 ELSE 12300
11780 NNC(SCN)=NNC(SCN)+TCS(SCN)
11790 JKL(SCN)=SCN
11800 GOTO 1170
11810 IF TT=TM GOTO 12100
11820 VS=VS+1
11830 IF CCD(VS)=1 GOTO 11820
11840 D(VS)=D(VS)-1
11850 NNC(VS)=NNC(VS)+TCS(VS)
11860 GOTO 1170
11870 IF TT=TM GOTO 11880 ELSE 11820
11880 JKL(VS)=VS
11890 IF VS=1 GOTO 12100
11900 FOR MP=1 TO VS-1
11910 IF CCD(MP)=1 GOTO 11940
11920 D(MP)=D(MP)+1
11930 NNC(MP)=NNC(MP)-TCS(MP)
11940 NEXT MP
11950 GB=1
11960 GOTO 1170
11970 IF TT=TM GOTO 12100
11980 FOR MP=1 TO VS-1
11990 IF CCD(MP)=1 GOTO 12090
12000 D(MP)=D(MP)-1
12010 NNC(MP)=NNC(MP)+TCS(MP)
12020 RM=1
12030 GOTO 1170
12040 IF TT=TM THEN VS=MP ELSE 12090
12050 RM=0
12060 JKL(MP)=MP
12070 IF VS=1 GOTO 12100
12080 GOTO 11900
12090 NEXT MP
12100 PTOTAL=0
12110 FOR QI=1 TO K
12120 PTOTAL=PTOTAL+TCS(JKL(QI))
12130 NEXT QI
12140 IF STOTAL<=PTOTAL GOTO 12190
12150 FOR QI=1 TO K
12160 IJ(QI)=JKL(QI)
12170 NEXT QI
12180 STOTAL=PTOTAL
12190 FOR QI=1 TO CZ
12200 IF JKL(QI)=0 GOTO 12230
12210 D(QI)=D(QI)+1
12220 NNC(QI)=NNC(QI)-TCS(QI)
12230 NEXT QI
12240 FOR QI=1 TO CZ
12250 JKL(QI)=0
12260 NEXT QI
12270 PTOTAL=0
```

```
12280 VS=0
12290 GB=0
12300 NEXT SCN
12310 JKL(IT)=0
12320 D(IT)=D(IT)+1
12330 NNC(IT)=NNC(IT)-TCS(IT)
12340 QJJ=0
12350 PD=1
12360 IC=0
12370 FOR TI=IT+1 TO K
12380 IF TCS(TI)=>STOTAL GOTO 12410
12390 IC=IC+1
12400 NEXT TI
12410 PI=IT+IC
12420 FOR MB=IT+1 TO PI
12430 IF D(MB)=CD(MB) THEN CCD(MB)=1
12440 IF D(MB)>CD(MB) THEN D(MB)=D(MB)-1 ELSE 12970
12450 NNC(MB)=NNC(MB)+TCS(MB)
12460 JI(MB)=MB
12470 GOTO 1170
12480 IF TT=TM GOTO 12770
12490 SV=SV+1
12500 IF CCD(SV)=1 GOTO 12490
12510 D(SV)=D(SV)-1
12520 NNC(SV)=NNC(SV)+TCS(SV)
12530 GOTO 1170
12540 IF TT=TM GOTO 12550 ELSE 12490
12550 JI(SV)=SV
12560 IF SV=1 GOTO 12770
12570 FOR JJ=1 TO SV-1
12580 IF CCD(JJ)=1 GOTO 12610
12590 D(JJ)=D(JJ)+1
12600 NNC(JJ)=NNC(JJ)-TCS(JJ)
12610 NEXT JJ
12620 ZL=1
12630 GOTO 1170
12640 IF TT=TM GOTO 12770
12650 FOR JJ=1 TO SV-1
12660 IF CCD(JJ)=1 GOTO 12760
12670 D(JJ)=D(JJ)-1
12680 NNC(JJ)=NNC(JJ)+TCS(JJ)
12690 XL=1
12700 GOTO 1170
12710 IF TT=TM THEN SV=JJ ELSE 12760
12720 XL=0
12730 JI(JJ)=JJ
12740 IF SV=1 GOTO 12770
12750 GOTO 12570
12760 NEXT JJ
12770 HTOTAL=0
12780 FOR QI=1 TO K
12790 HTOTAL=HTOTAL+TCS(JI(QI))
```

```
12800 NEXT QI
12810 IF STOTAL<=HTOTAL GOTO 12860
12820 FOR QI=1 TO K
12830 IJ(QI)=JI(QI)
12840 NEXT QI
12850 STOTAL=HTOTAL
12860 FOR QI=1 TO K
12870 IF JI(QI)=0 GOTO 12900
12880 D(QI)=D(QI)+1
12890 NNC(QI)=NNC(QI)-TCS(QI)
12900 NEXT QI
12910 FOR QI=1 TO K
12920 JI(QI)=0
12930 NEXT QI
12940 HTOTAL=0
12950 SV=0
12960 ZL=0
12970 NEXT MB
12980 FOR QI=1 TO K
12990 IF IJ(QI)=0 GOTO 13020
13000 D(QI)=D(QI)-1
13010 NNC(QI)=NNC(QI)+TCS(QI)
13020 NEXT QI
13030 SP=SP+1
13040 SPP=SPP+SP
13050 GOTO 1170
13060 PD=0
13070 ED=0
13080 YK=0
13090 XK=0
13100 VP=0
13110 SP=0
13120 XB=0
13130 GOTO 9940
13140 REM
13150 REM      FINISHING ITERATIONS AND PRINTING OF GRAPHS AT THE CRASH POINT
13160 REM
13170 IF WAW$<>"Y" GOTO 13220
13180 DEVICE$="LPT1:":WIDTH "LPT1:",80
13190 OPEN DEVICE$ FOR OUTPUT AS #1
13200 PRINT #1, CHR$(12)
13210 CLOSE #1
13220 PRINT " "
13230 IF WAW$="Y" GOTO 13390
13240 INPUT "DO YOU WISH TO GET A HARD COPY OF THIS TABLE (Y OR N)";WAW$
13250 CLS
13260 SPP=0
13270 IF WAW$<>"Y" GOTO 13450
13280 PRINT "THE OUTPUT IS GOING TO THE PRINTER / IT MAY BE SLOW"
13290 FOR I=1 TO N
13300 D(I)=XD(I)
13310 NNC(I)=NCT(I)
```

```
13320 CCD(I)=0
13330 NEXT I
13340 DIRC=DRC
13350 CHANGE=DRC
13360 TOTCOST=0
13370 XB=0
13380 GOTO 1170
13390 CLS
13400 INPUT "DO YOU WISH TO GET ANOTHER HARD COPY OF THE TABLE (Y OR N)";GG$
13410 CLS
13420 IF GG$<>"Y" GOTO 13450
13430 SPP=0
13440 GOTO 13280
13450 GH=1
13460 FOR II=1 TO K
13470 IF CCD(II)=1 GOTO 13500
13480 D(II)=D(II)+1
13490 NNC(II)=NNC(II)-TCS(II)
13500 NEXT II
13510 GOTO 1170
13520 REM
13530 REM      TABLE OF COMPRESSION ITERATIONS
13540 REM
13550 IF WAW$="Y" GOTO 13580
13560 DEVICE$="SCRN:":WIDTH "SCRN:",80
13570 GOTO 13590
13580 DEVICE$="LPT1:":WIDTH "LPT1:",80
13590 OPEN DEVICE$ FOR OUTPUT AS #1
13600 IF SPP>1 GOTO 13730
13610 PRINT #1, TAB(28);"COMPRESSION COST ITERATIONS"
13620 PRINT #1, " "
13630 PRINT #1, " "
13640 PRINT #1, TAB(3);"NORMAL PROJECT DURATION:";T1;TAB(43);"NORMAL PROJECT DIR
ECT COST: $";DRC
13650 IF ICD$<>"Y" GOTO 13670
13660 PRINT #1, TAB(3);"NORMAL PROJECT INDIRECT COST: $";NINDCT;TAB(43);"NORMAL
PROJECT TOTAL COST:  $";DRC+NINDCT
13670 PRINT #1, " "
13680 PRINT #1, " "
13690 IF ICD$<>"Y" GOTO 13760
13700 PRINT #1, TAB(3);"COMPRESSED";TAB(16);"INCREASED";TAB(30);"PROJECT";TAB(44
);"PROJECT";TAB(57);"PROJECT";TAB(70);"PROJECT"
13710 PRINT #1, TAB(3);"ACTIVITY(S)";TAB(16);"DIRECT COSTS";TAB(30);"DIRECT COST
S";TAB(44);"INDIRECT CT";TAB(57);"TOTAL COST";TAB(70);"DURATION"
13720 PRINT #1, TAB(3);"-----------";TAB(16);"------------";TAB(30);"-----------
-";TAB(44);"-----------";TAB(57);"----------";TAB(70);"--------"
13730 IF ICD$<>"Y" GOTO 13790
13740 PRINT #1, TAB(5)
13750 GOTO 13800
13760 PRINT #1, TAB(10);"COMPRESSED";TAB(27);"INCREASED";TAB(43);"PROJECT";TAB(5
8);"PROJECT"
13770 PRINT #1, TAB(10);"ACTIVITY(S)";TAB(26);"DIRECT COSTS";TAB(42);"DIRECT COS
TS";TAB(58);"DURATION"
```

```
13780 PRINT #1, TAB(10);"-----------";TAB(26);"------------";TAB(42);"----------
--";TAB(58);"--------"
13790 PRINT #1, TAB(12);
13800 FOR I=1 TO N
13810 HH(I)=0
13820 NEXT I
13830 FOR I=1 TO N
13840 IF IJ(I)=0 GOTO 13890
13850 PRINT #1, A$(I);
13860 HH(I)=1
13870 VB$=A$(I)
13880 GOTO 13900
13890 NEXT I
13900 FOR I=1 TO N
13910 IF IJ(I)=0 GOTO 13960
13920 IF A$(I)=VB$ GOTO 13960
13930 PRINT #1, ",";A$(I);
13940 VB$=A$(I)
13950 HH(I)=1
13960 NEXT I
13970 OLDCOST=TOTCOST
13980 TOTCOST=DIRC+INDCT
13990 IF ICD$<>"Y" GOTO 14140
14000 PRINT #1, TAB(19);INCR;TAB(32);DIRC;TAB(46);INDCT;TAB(58);TOTCOST;TAB(71);
TT
14010 IF OLDCOST=0 GOTO 14130
14020 IF TOTCOST<=OLDCOST GOTO 14130
14030 IF TOPT=1 GOTO 14130
14040 FOR I=1 TO N
14050 IF HH(I)=0 GOTO 14090
14060 DPT(I)=D(I)+1
14070 NPT(I)=NNC(I)-TCS(I)
14080 GOTO 14110
14090 DPT(I)=D(I)
14100 NPT(I)=NNC(I)
14110 NEXT I
14120 TOPT=1
14130 GOTO 14150
14140 PRINT #1, TAB(29);INCR;TAB(43);DIRC;TAB(60);TT
14150 CLOSE #1
14160 GOTO 13060
14170 REM
14180 REM      RECALCULATE FOR THE OPTIMUM PROJECT DURATION
14190 REM
14200 CLS
14210 IF ICD$<>"Y" GOTO 14340
14220 IF TOPT=1 GOTO 14260
14230 OPTCOST=1
14240 GH=0
14250 GOTO 5470
14260 OPTCOST=1
14270 GH=0
```

```
14280 FOR I=1 TO N
14290 D(I)=DPT(I)
14300 NNC(I)=NPT(I)
14310 NEXT I
14320 CLS
14330 GOTO 1170
14340 END
```

APPENDIX B
SAMPLE REPORTS PRODUCED BY CRASH

THE FOLLOWING DATA IS RECORDED ON THE PROJECT:

PROJECT NAME: ECI 5147 HOMEWORK
SCHEDULED BY: MARK A RONCOLI
DATE ESTIMATED: 17 NOV 86

ACT. CODE	ACTIVITY DESCRIPTION	DURATION NORM/CRASH	ACT. COST NORM/CRASH	DEPENDENCIES
A		6.0/ 4.0	650/ 700	-
C		6.0/ 5.0	730/ 755	-
D		4.0/ 3.0	180/ 225	-
E		5.0/ 2.0	485/ 590	D
B		4.0/ 3.0	260/ 305	A
F		3.0/ 1.0	300/ 340	C, E
G		8.0/ 6.0	540/ 600	B, F
H		3.0/ 2.0	190/ 230	F
I		5.0/ 3.0	385/ 455	F
J		6.0/ 4.0	620/ 680	I
K		7.0/ 5.0	510/ 550	H, I
N		6.0/ 4.0	800/ 890	G
L		2.0/ 1.0	170/ 280	J
M		3.0/ 2.0	100/ 135	K, L

PROJECT INDIRECT COST= $ 95 PER DAY

NORMAL COMPLETION TIME
CPM SCHEDULING

PROJECT: ECI 5147 HOMEWORK
TIME SCALE: DAYS

ACT. CODE	ACTIVITY DESCRIPTION	ACT. DUR.	EARLY START	EARLY FINISH	LATEST START	LATEST FINISH	TOTAL FLOAT	CP
A		6.0	0.0	6.0	4.0	10.0	4.0	-
C		6.0	0.0	6.0	3.0	9.0	3.0	-
D		4.0	0.0	4.0	0.0	4.0	0.0	C
E		5.0	4.0	9.0	4.0	9.0	0.0	C
B		4.0	6.0	10.0	10.0	14.0	4.0	-
F		3.0	9.0	12.0	9.0	12.0	0.0	C
G		8.0	12.0	20.0	14.0	22.0	2.0	-
H		3.0	12.0	15.0	15.0	18.0	3.0	-
I		5.0	12.0	17.0	12.0	17.0	0.0	C
J		6.0	17.0	23.0	17.0	23.0	0.0	C
K		7.0	17.0	24.0	18.0	25.0	1.0	-
N		6.0	20.0	26.0	22.0	28.0	2.0	-
L		2.0	23.0	25.0	23.0	25.0	0.0	C
M		3.0	25.0	28.0	25.0	28.0	0.0	C

PROJECT DURATION: 28

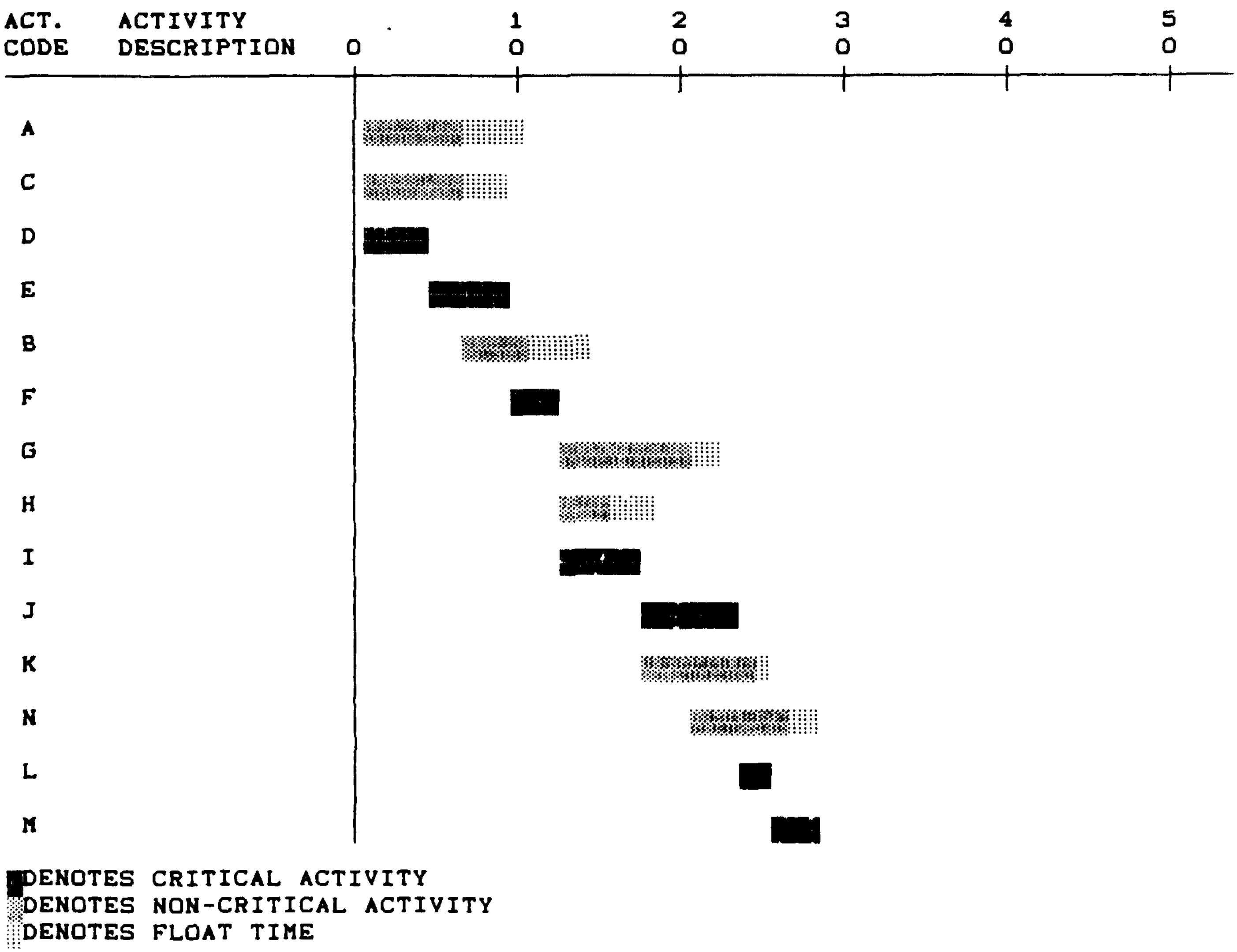
PROJECT DIRECT COST: $ 5920

PROJECT INDIRECT COST: $ 2660

PROJECT TOTAL COST: $ 8580

NORMAL COMPLETION TIME
BAR (GANNT) CHART

PROJECT: ECI 5147 HOMEWORK
TIME SCALE: DAYS



COMPRESSION COST ITERATIONS

NORMAL PROJECT DURATION: 28 NORMAL PROJECT DIRECT COST: $ 5920
NORMAL PROJECT INDIRECT COST: $ 2660 NORMAL PROJECT TOTAL COST: $ 8580

COMPRESSED ACTIVITY(S)	INCREASED DIRECT COSTS	PROJECT DIRECT COSTS	PROJECT INDIRECT CT	PROJECT TOTAL COST	PROJECT DURATION
F	20	5940	2565	8505	27
F	20	5960	2470	8430	26
J	30	5990	2375	8365	25
E	35	6025	2280	8305	24
A, E	60	6085	2185	8270	23
A, E	60	6145	2090	8235	22
I, G	65	6210	1995	8205	21
I, G	65	6275	1900	8175	20
B, M	80	6355	1805	8160	19
J, K, N	95	6450	1710	8160	18
C, D, N	115	6565	1615	8180	17

CRASHED COMPLETION TIME
CPM SCHEDULING

PROJECT: ECI 5147 HOMEWORK
TIME SCALE: DAYS

ACT. CODE	ACTIVITY DESCRIPTION	ACT. DUR.	EARLY START	EARLY FINISH	LATEST START	LATEST FINISH	TOTAL FLOAT	CF
C		5.0	0.0	5.0	0.0	5.0	0.0	C
D		3.0	0.0	3.0	0.0	3.0	0.0	C
A		4.0	0.0	4.0	0.0	4.0	0.0	C
E		2.0	3.0	5.0	3.0	5.0	0.0	C
B		3.0	4.0	7.0	4.0	7.0	0.0	C
F		1.0	5.0	6.0	5.0	6.0	0.0	C
H		3.0	6.0	9.0	6.0	9.0	0.0	C
I		3.0	6.0	9.0	6.0	9.0	0.0	C
G		6.0	7.0	13.0	7.0	13.0	0.0	C
J		4.0	9.0	13.0	9.0	13.0	0.0	C
K		6.0	9.0	15.0	9.0	15.0	0.0	C
L		2.0	13.0	15.0	13.0	15.0	0.0	C
N		4.0	13.0	17.0	13.0	17.0	0.0	C
M		2.0	15.0	17.0	15.0	17.0	0.0	C

PROJECT DURATION: 17

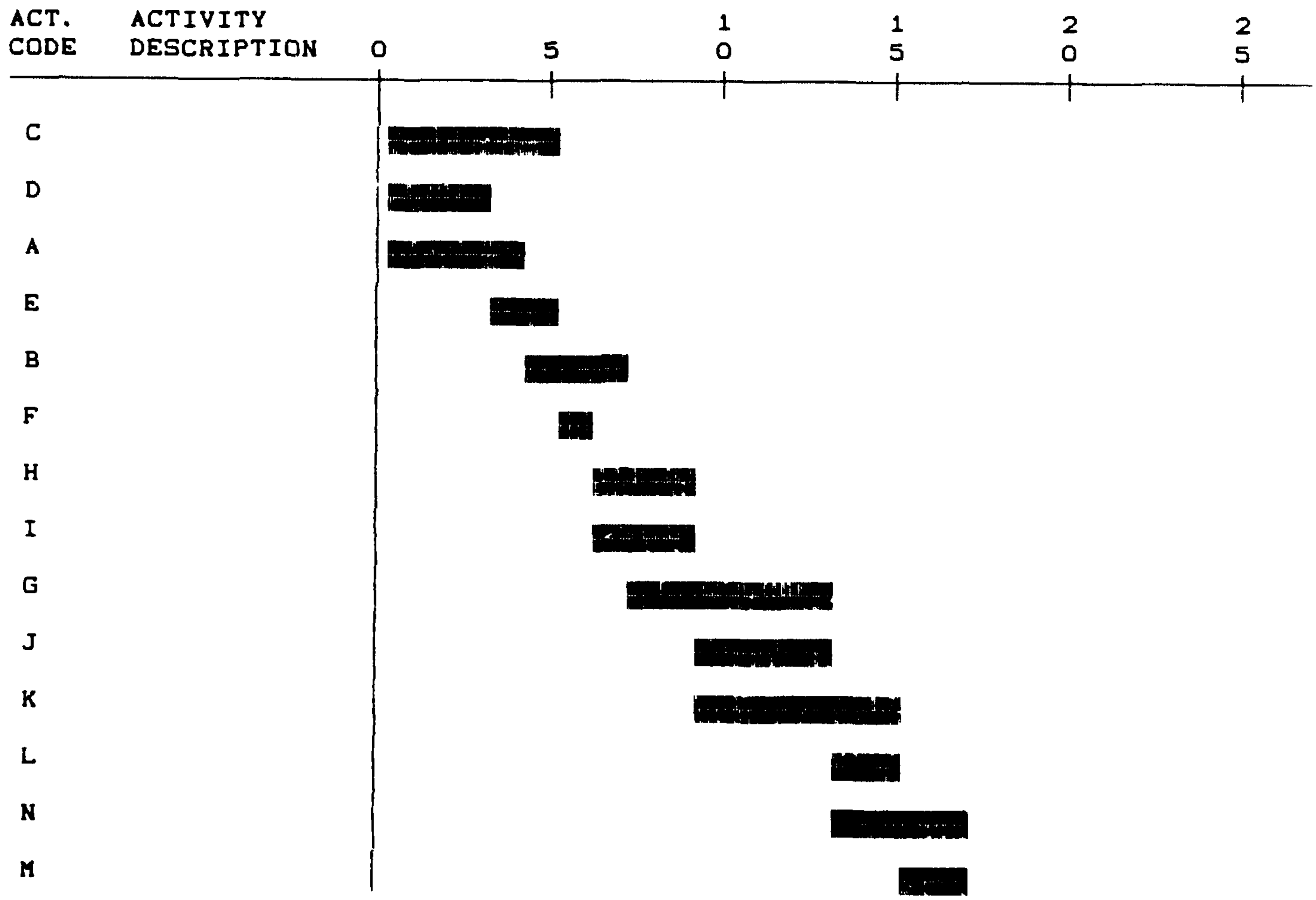
PROJECT DIRECT COST: $ 6565

PROJECT INDIRECT COST: $ 1615

PROJECT TOTAL COST: $ 8180

CRASHED COMPLETION TIME
BAR (GANNT) CHART

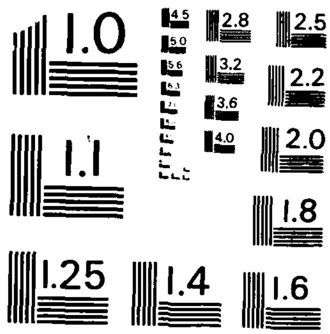
PROJECT: ECI 5147 HOMEWORK
TIME SCALE: DAYS



DENOTES CRITICAL ACTIVITY
DENOTES NON-CRITICAL ACTIVITY
DENOTES FLOAT TIME

AD-A175 137 COMPUTERIZED COST OPTIMIZATION SCHEDULING(U) ARMY MILITARY PERSONNEL CENTER ALEXANDRIA VA M A RONCOLI 21 NOV 86 2/2

UNCLASSIFIED F/G 14/1 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

OPTIMUM COST COMPLETION TIME
CPM SCHEDULING

PROJECT: ECI 5147 HOMEWORK
TIME SCALE: DAYS

ACT. CODE	ACTIVITY DESCRIPTION	ACT. DUR.	EARLY START	EARLY FINISH	LATEST START	LATEST FINISH	TOTAL FLOAT	CF
C		6.0	0.0	6.0	0.0	6.0	0.0	C
D		4.0	0.0	4.0	0.0	4.0	0.0	C
A		4.0	0.0	4.0	0.0	4.0	0.0	C
E		2.0	4.0	6.0	4.0	6.0	0.0	C
B		3.0	4.0	7.0	4.0	7.0	0.0	C
F		1.0	6.0	7.0	6.0	7.0	0.0	C
H		3.0	7.0	10.0	7.0	10.0	0.0	C
I		3.0	7.0	10.0	7.0	10.0	0.0	C
G		6.0	7.0	13.0	7.0	13.0	0.0	C
J		4.0	10.0	14.0	10.0	14.0	0.0	C
K		6.0	10.0	16.0	10.0	16.0	0.0	C
N		5.0	13.0	18.0	13.0	18.0	0.0	C
L		2.0	14.0	16.0	14.0	16.0	0.0	C
M		2.0	16.0	18.0	16.0	18.0	0.0	C

PROJECT DURATION: 18

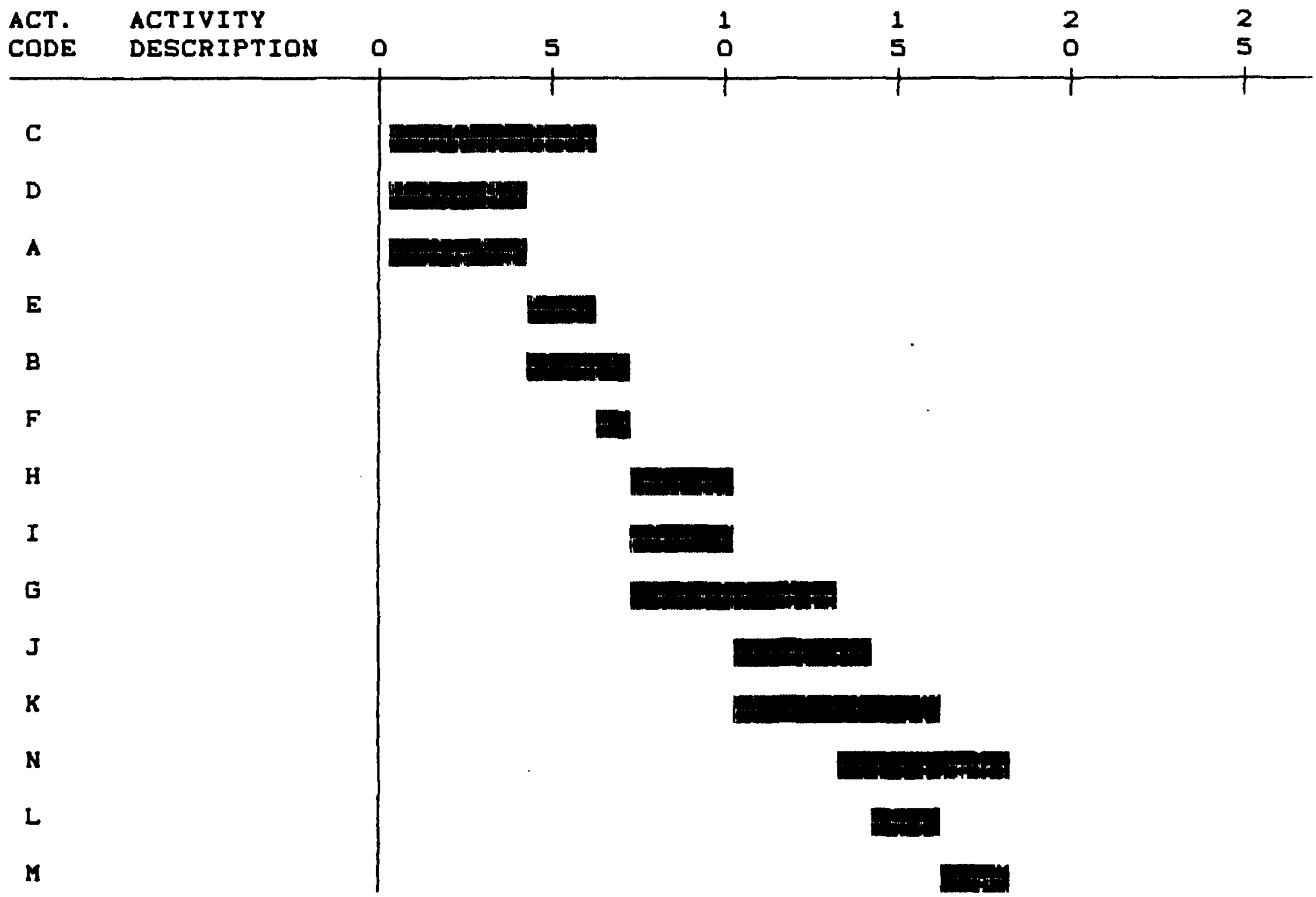
PROJECT DIRECT COST: $ 6450

PROJECT INDIRECT COST: $ 1710

PROJECT TOTAL COST: $ 8160

OPTIMUM COST COMPLETION TIME
BAR (GANNT) CHART

PROJECT: ECI 5147 HOMEWORK
TIME SCALE: DAYS



DENOTES CRITICAL ACTIVITY
DENOTES NON-CRITICAL ACTIVITY
DENOTES FLOAT TIME

REFERENCES

1. Adrian, James J., Microcomputers in the Construction Industry, Reston Publishing Company, Reston, Va, 1985.

2. Antill, James M., and Ronald W. Woodread, Critical Path Methods in Construction Practice, 2d ed., John Wiley and Sons, Inc., New York, 1970.

3. Apple Computers, The Macproject Manual for MacIntosh, 1985.

4. Burman, Peter J., Precedence Networks for Project Planning and Control, McGraw-Hill Book Company, London, 1972.

5. Clough, Richard H., Construction Project Management, John Wiley and Sons, Inc., New York, 1972.

6. Harris, Robert B., Precedence and Arrow Networking Techniques for Construction, John Wiley and Sons, New York, 1978.

7. Horowitz, Joseph, Critical Path Scheduling, The Ronald Press Company, New York, 1967.

8. Moder, Joseph J., and Cecil R. Phillips, Project Management with CPM and PERT, 2d ed., Van Nostrand Reinhold Co., New York, 1970.

9. O'Brien, James J. CPM in Construction Management, 3rd ed., McGraw-Hill Book Company, New York, 1984.

10. Park, William R., Construction Bidding For Profit, John Wiley and Sons, Inc., New York, 1979.

11. US Army Engineer School, Student Workbook: Engineer Management Advanced Applications, Fort Belvoir, Va, 1981.

12. Willis, Edward M., Scheduling Construction Projects, John Wiley and Sons, Inc., New York, 1986.

END

2-81

DTIC